Working with the IBM Grid Toolbox
Monterey Grid Computing Conference

August 12, 2004

Duane Quintern
Chris Dawson
Michael Osias

© 2004 IBM Corporation

Agenda Summary

Module	Content Owner	Module Time
•Module 1 – Introduction	Mike	8:00 15 minutes
•Module 2 – Getting Started	Duane	8:15 25 minutes
•Module 3 – Installing IGT	Duane	8:40 1 hour
•Break		9:40 15 minutes
•Module 4 – Administration and Dev	Mike	9:55 30 minutes
•Module 5 – Extensions to IGT	Chris	10:25 1 hour
•Module 6 – Customer Examples	Chris	11:25 30 minutes
•Homework assignment	All	11:55 15 minutes
	Total	12:10 Adjourn

© 2004 IBM Corporation

Agenda

- **Module 1 – Introduction**
  - **What is grid (value, etc)**
  - **What is the IBM Grid Toolbox**
  - **What you will learn**
- **Module 2 – Getting started**
  - **Environment overview**
  - **Pre-requisites**
  - **Tips and tricks**
- **Module 3 – Installing IGT**
  - **Running the installer**
  - **Post installation configuration tasks**

© 2004 IBM Corporation

Agenda

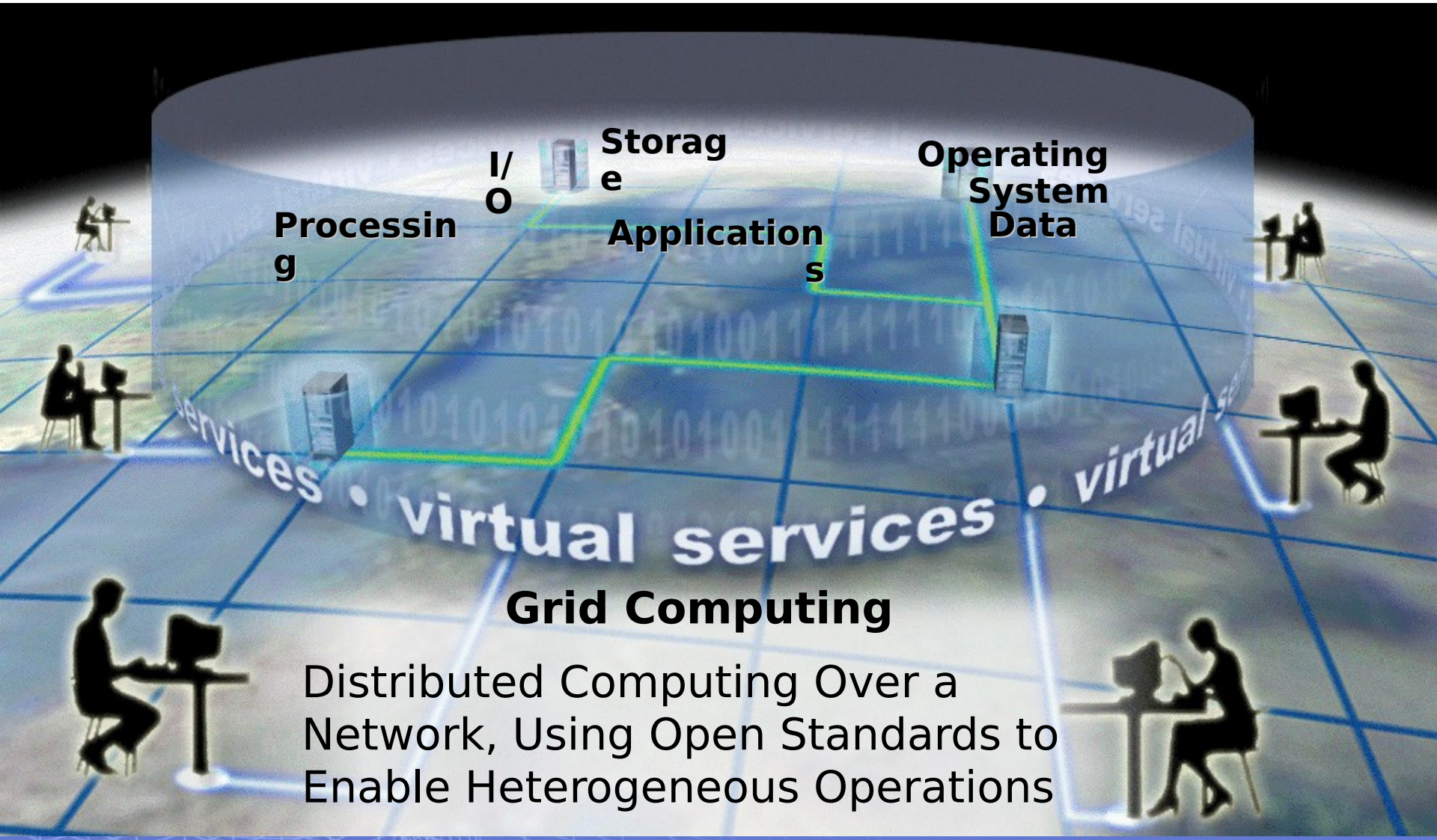
- **Module 3 – Installing IGT (cont.)**
  - **Configuring security**
  - **Testing the installation**
- **Module 4 - Administration and development**
  - **Using the Grid Services web interface**
  - **Overview of GARs**
  - **Tools to build grid services**
- **Module 5 - Extension to IGT**
  - **LSF, Condor**
- **Module 6 – Customer Examples**
- **Homework assignment**

© 2004 IBM Corporation

Module 1 - Introduction

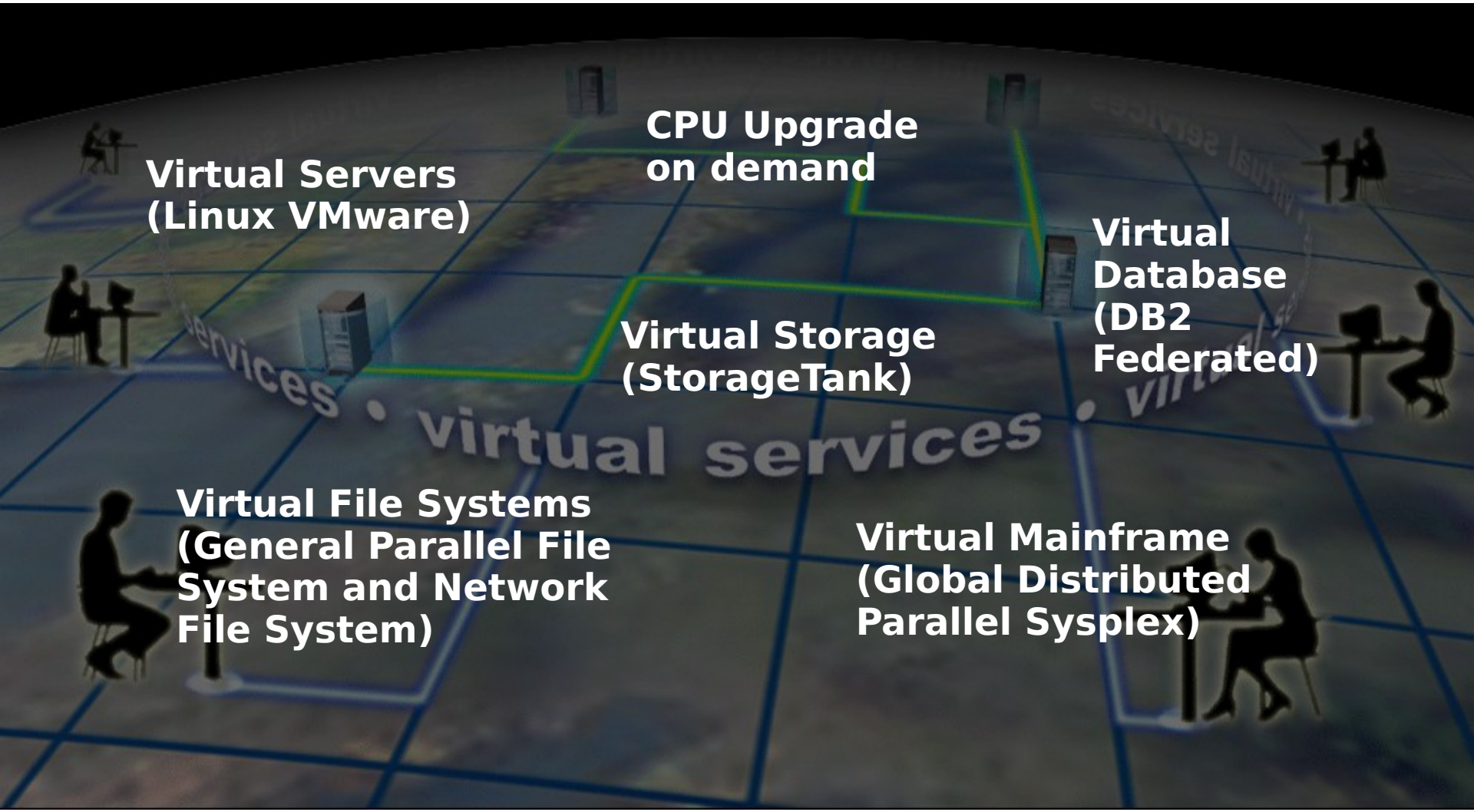
© 2004 IBM Corporation

Virtualized



© 2004 IBM Corporation

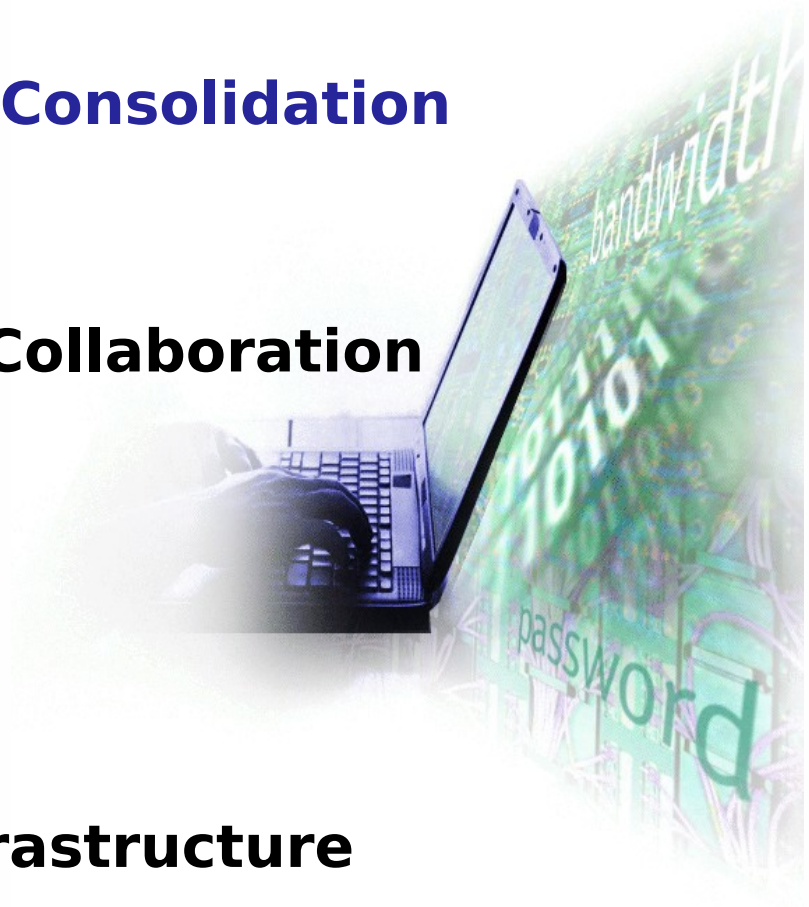
Virtualization Technologies Today



© 2004 IBM Corporation

Grid Addresses These Needs

- **Infrastructure Optimization**
  - **Workload Management and Consolidation**
  - **Reduced Cycle Times**
- **Increased Access to Data and Collaboration**
  - **Virtual organizations and resource sharing**
  - **Federation of Data**
  - **Global Distribution**
- **Resilient / Highly Available Infrastructure**
  - **Business Continuity**
  - **Recovery and Failover**



© 2004 IBM Corporation

The Value of Open Standards

Distributed Computing:
Grid
(Globus -> OGSA)

Applications:
Web Services
(SOAP, WSDL, UDDI)

Operating System:
Linux

Information:
World-wide Web
(html, http, j2ee, xml)

Communications:
e-mail
(pop3,SMTP,Mime)

Networking:
The Internet
(TCP/IP)

e-business

© 2004 IBM Corporation

Global Grid Forum

- Analogous to W3C
- Working and research groups
- Ties to WS-I, IETF, W3C, etc.
- Broad participation
- Industry, Academia, Science & Research

Industry Participation

IBM Platinum sponsor, actively involved in many WG

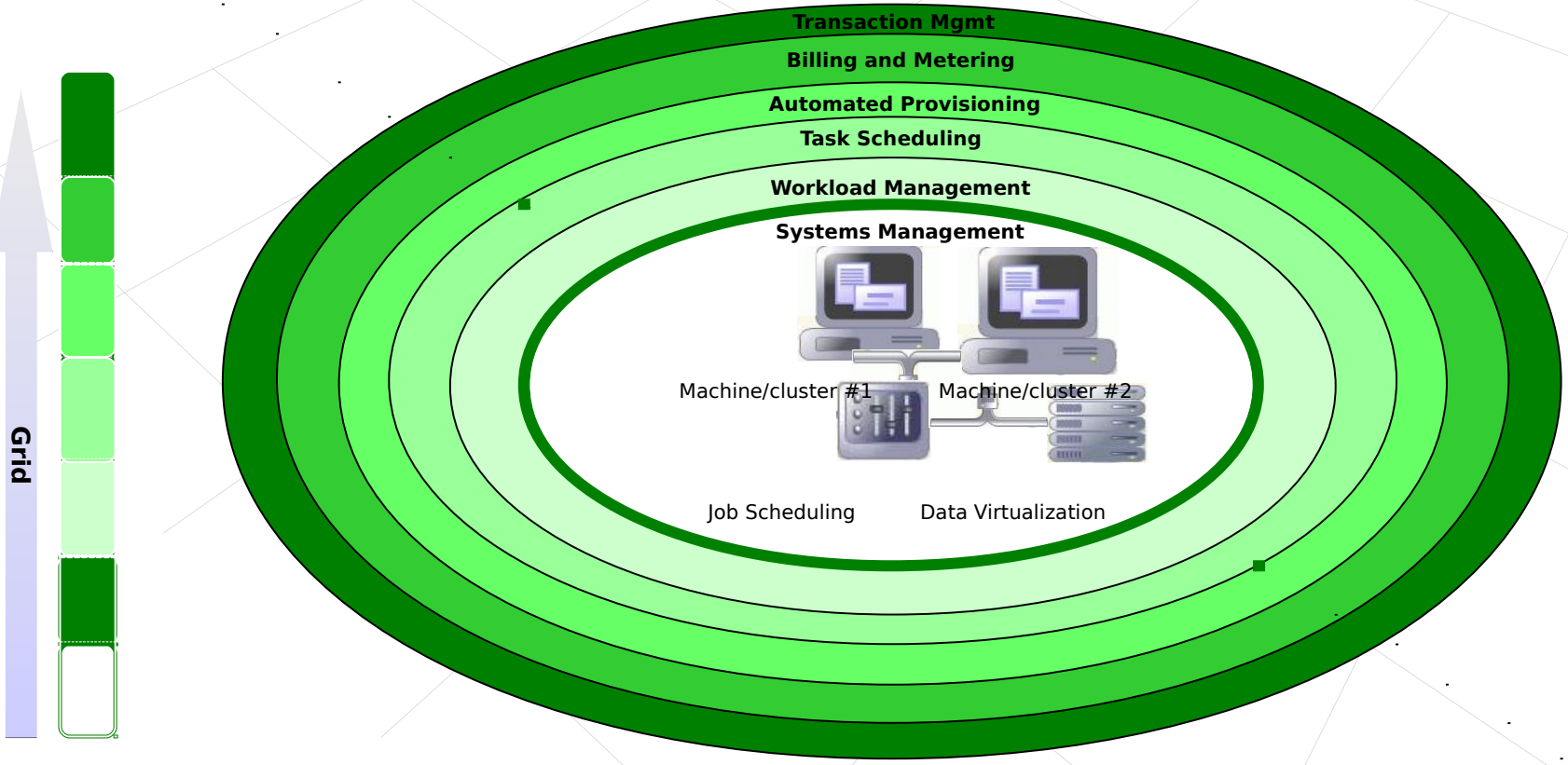
Sun microsystems, hp, ORACLE, EDS, bea, DELL, Microsoft

© 2004 IBM Corporation

What Makes Up a Grid

Transaction Management:

- **Manage the execution of e-business transactions across distributed resources**
- **Enable dynamic allocation of resources for transactional and parallel application models**



On Demand Roadmap

Transaction Management

WebSphere.

Billing and Metering

Tivoli.

Workload Management

@server Tivoli.

Information Virtualization

@server TotalStorage™ DB2. Tivoli. AVAKI

Orchestrated Provisioning

@server Tivoli.

Scheduling

WebSphere. Tivoli. DataSynapse Platform UNITED DEVICES™ GridXpert

Web Services/OGSA

@server the globus alliance WebSphere.

Open Grid Services Architecture (OGSA) extends and enhances web services with Grid capabilities.

- OGSA Introduces extensions to web services
  - Autonomic capabilities

  Provisioning

  Introduces state management to web services

  Reliable, secure management of distributed state

Lines between applications and resources are blurring

OGSA extends key middleware as meta-os services

Connectors and adapters virtualize native resources

Logical rendering of physical resources

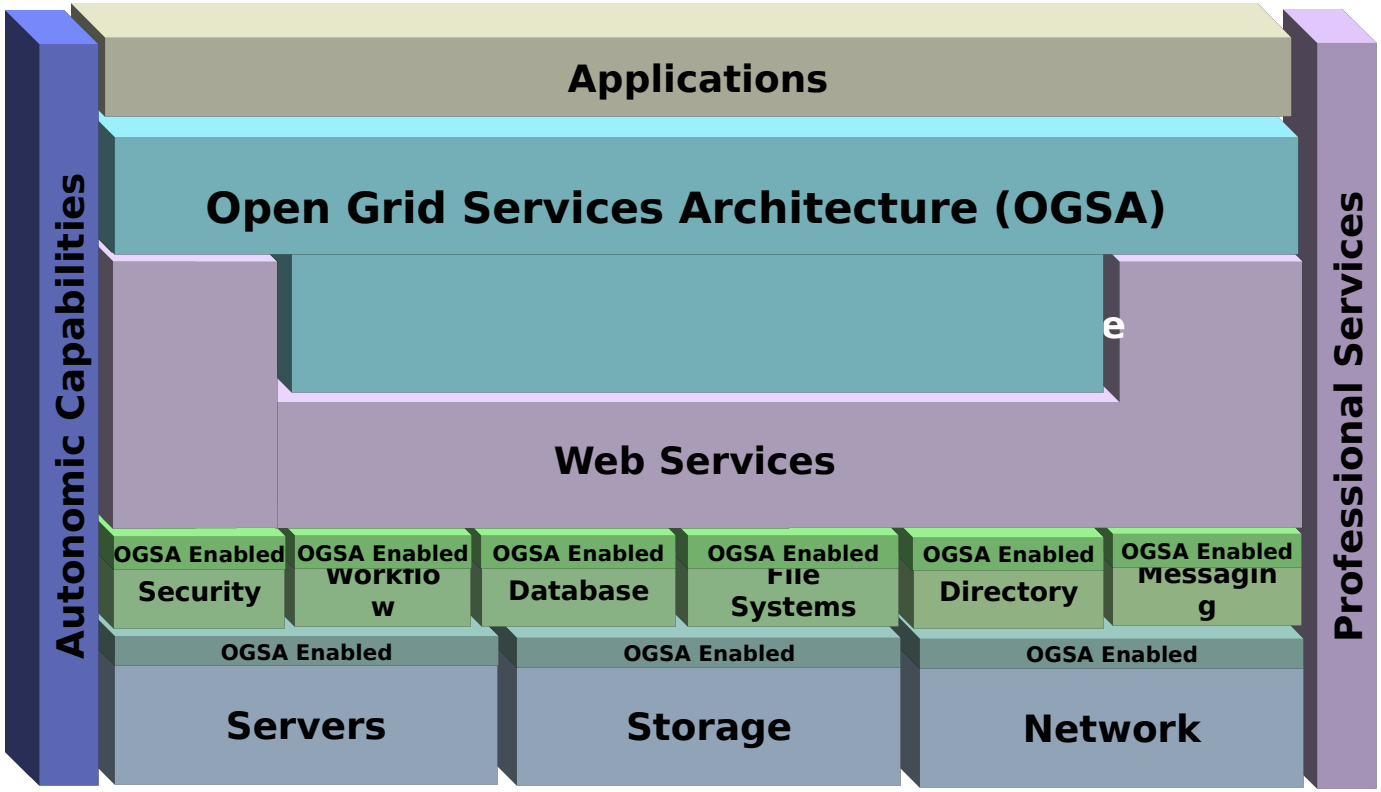
Service compensation layer

Resources rendered as services

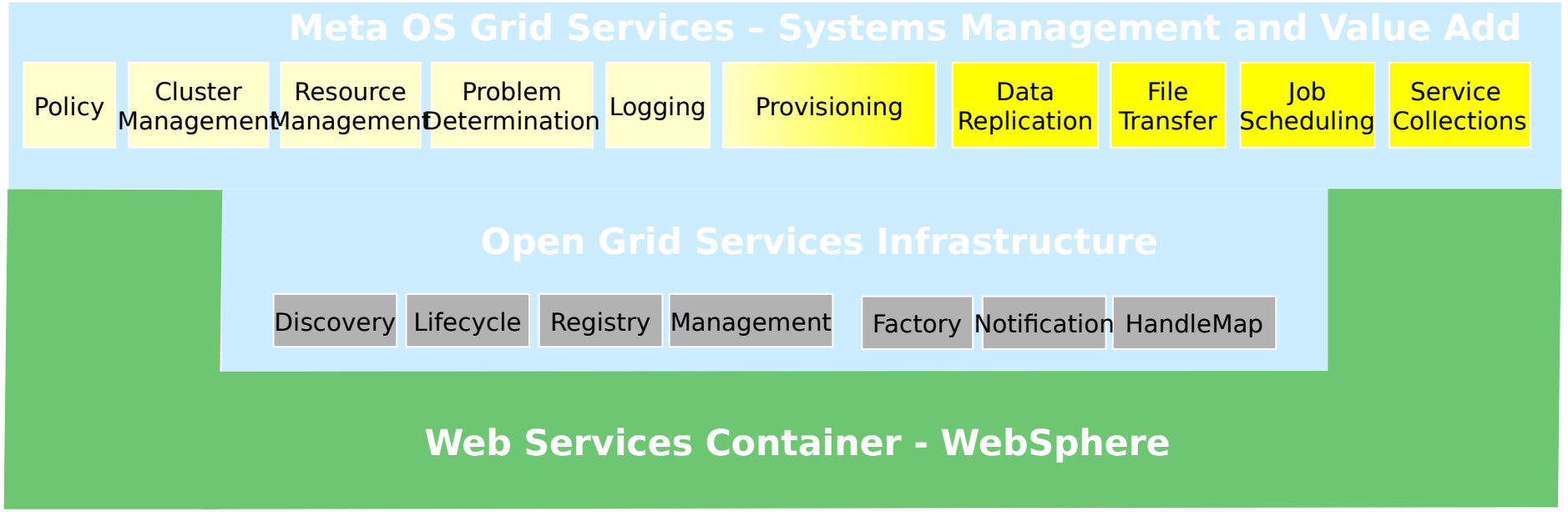
Separate interface from access and implementation

OGSA provides Grid Programming Model Standardization

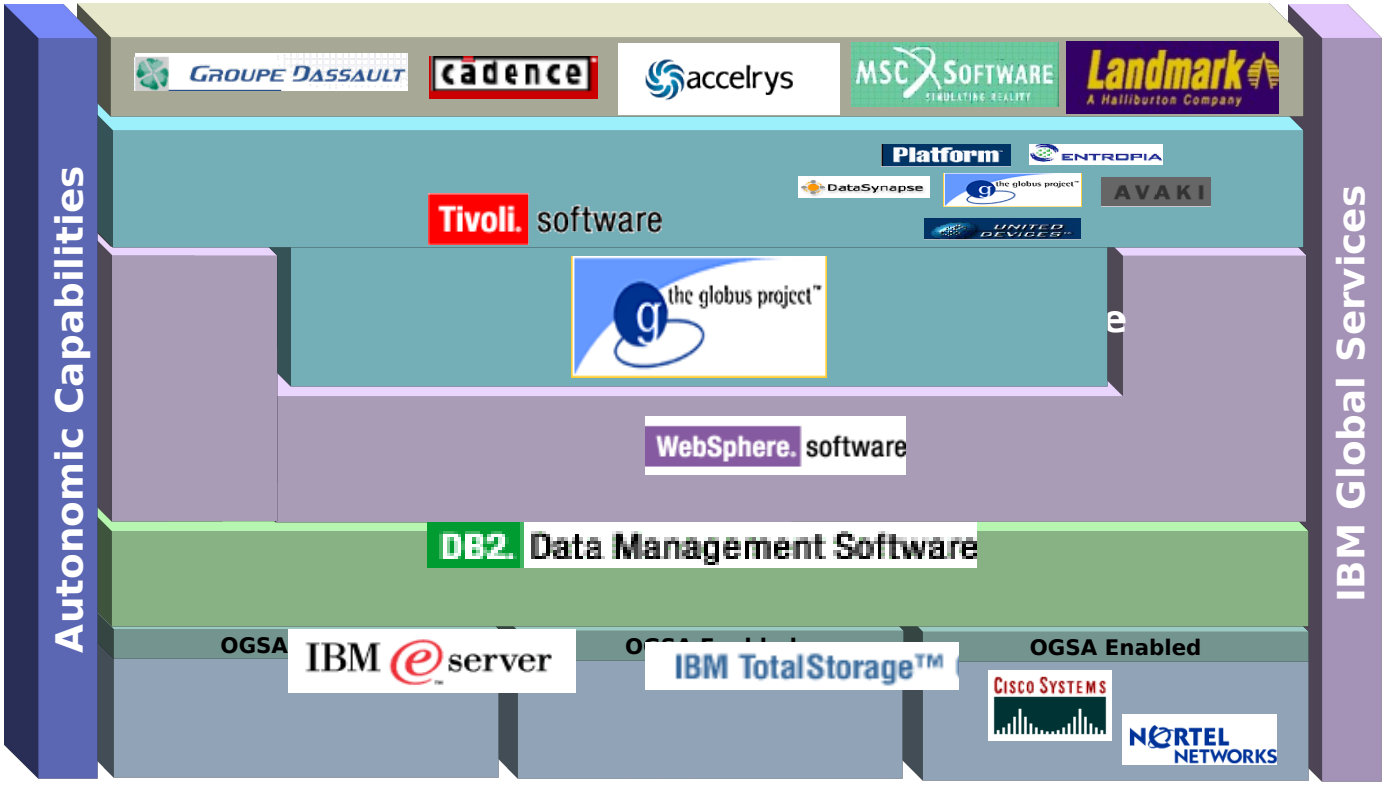
Architecture Framework



There are several key infrastructure and meta-os services that are part of OGSA and are fundamental to a well formed Grid architecture.



Architecture Framework



IBM Grid Toolbox for Multiplatforms V3.0

- A commercial derivative of the Globus Toolkit 3.0 with IBM Value Add
  - A platform for the **Grid Developer** to develop and test grid service and grid applications.
  - A platform for the **Grid Builder/Deployer** to host grid service and grid applications.

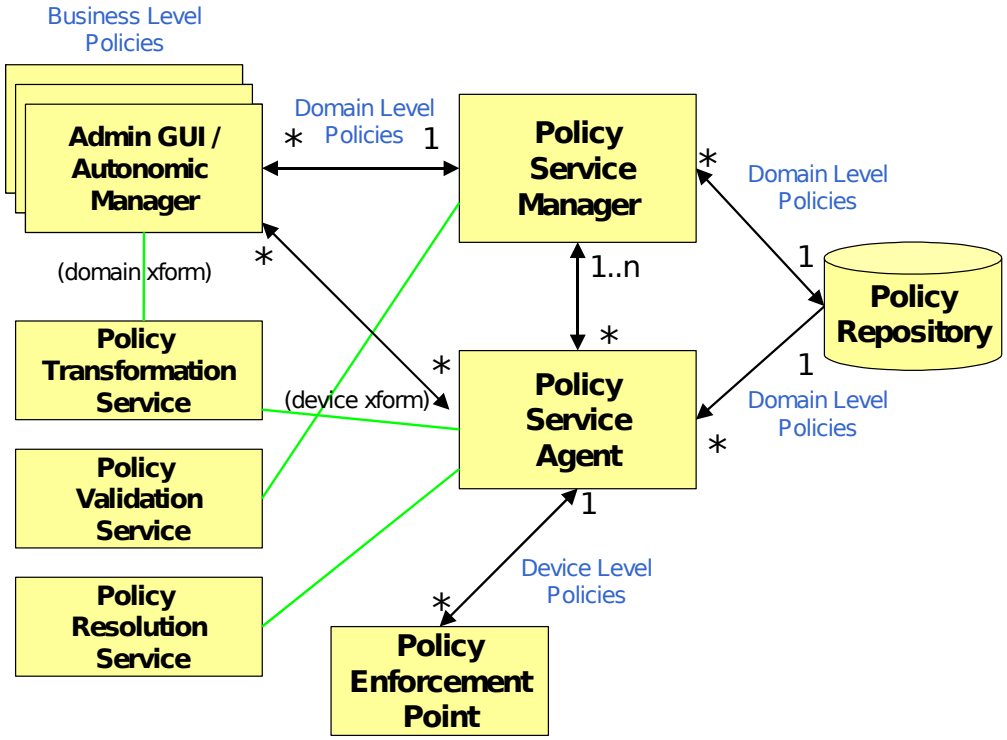
stomer scenarios used to design, document and test

IBM Grid Toolbox– the details....

- Integrated w**izard based installation**
- A **grid service runtime** environment based on the GGF Grid Service (OGSI) specification for hosting grid services.
- A web-based **management application** used to manage services within the runtime environment.
- A digital certificate base grid security infrastructure (**GSI**).
- Complete set of **configuration** and **administration** commands.
- **Development tools** to build, package and use grid services.
- Common (base) **grid services** for...
  - Discovery via Service Group
  - Policy Management
  - Common Management Models (CMM)
  - GT3.0 Program Execution Service (GT3.0 GRAM - MMJFS, MJS, UHE)
  - GT3.0 Information Services (GT3.0 MDS - Index)
  - GT3.0 Data Management Services (Multi-RFT)
- **Information Center** including tutorials to assist with the education and understanding of the technologies and capabilities packaged within the product.
  - **Sample** grid services and applications demonstrating key capabilities.

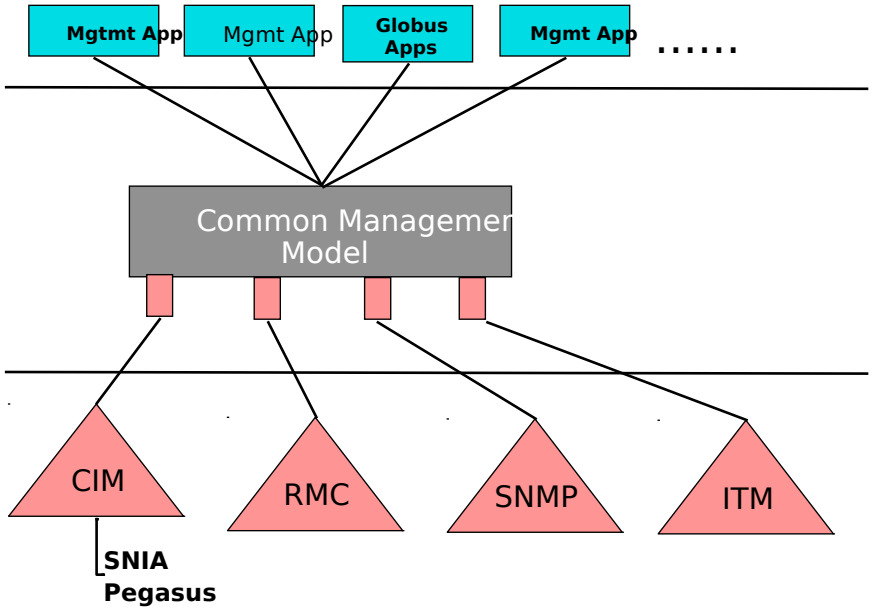
Common Policy Management Services

- Common policy service manager (PSM) and policy service agent (PSA).
  - Includes a sample Network QoS discipline.
- PSM and PSA provide common components required for any policy based application or solution.
- PSM allows policies to be generically..
  - Added
  - Updated
  - Removed
  - Discovered (found)
- PSA generically…
  - Activates/Deactivates policies
  - Manages active policies
- Repository implemented using embedded database.



Common Management Models - CMM

- Exposes systems resources as services
  - System, filesystem, os, etc.
- Defines and models resource associations with service data
  - Hosted filesystems, running os, etc.
- Provides mapping layer (abstraction) between existing systems management systems and technologies (i.e. CIM, SNMP)
- JCA used to manage interactions with instrumentation.
  - Adapter needed per instrumentation type.
- Initial release includes a JCA Adapter for CIM allowing interactions with Pegasus CIMOM.



What you will learn

- How to install the IBM Grid Toolbox (IGT)
- Security and configuration
- Administration and development
- Extending IGT
- Example Projects

Module 2 – Getting Started

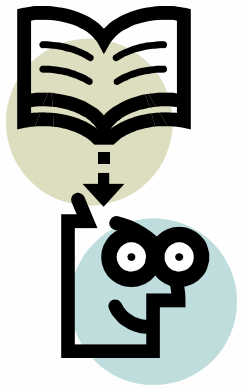
Before you install IGT…

AIX	V5.2 with 5200-01	1GB RAM 967 MB /opt 400 MB /tmp
Linux on xSeries	Redhat Enterprise Linux Advanced Server for Intel 2.1 SUSE LINUX Enterprise Server 8.0	1GB RAM 2Ghz processor 500 MB /opt 250 MB for install image 400 MB /tmp
Linux on zSeries	SUSE LINUX Enterprise Server 8.0 (31bit) Service Pack 2	512 MB RAM 500 MB /opt 250 MB for install image 400 MB /tmp
Linux on iSeries or pSeries	SUSE LINUX Enterprise Server 8.0 – Powered by UnitedLinux v1.0	512 MB RAM 500 MB /opt
OS/400	V5R3	TCP/IP 5722-TC1 Java 5722-JV1 WAS 5722IWE QShell 5722-SS1 opt 30 Host Svrs 5722-SS1 opt 12

Optional software

- ANT 1.5.x
  - http://.ant.apache.org
- Java 1.3.1 (SDK required if developing java application
  - http//java.sun.com

Module 3 – Installing IGT



© 2004 IBM Corporation

IGT Installation Steps

As root:

1. Install ANT: (1.5.1)

1.1 Untar ANT into /opt

1.2 In /etc/profile add

1.3 ANT_HOME=/opt/jakarta-ant-1.5.1

1.4 pathmunge $ANT_HOME/bin after

or

PATH=$PATH:$ANT_HOME/bin

1.5 export ANT_HOME PATH

2. Configure host information

2.1 Ensure the IP and the DNS hostname are the first entry in /etc/hosts, or at least come before:

127.0.0.1 localhost.localdomain

i.e.

192.168.0.30 mygridhost.epa.com

127.0.0.1 localhost localhost.localdomain

2.4 Ensure other host information is specified:

/etc/hostname

/proc/sys/kernel/hostname

2.5 Ensure /etc/resolv.conf is configured properly

© 2004 IBM Corporation

IGT Installation Steps (cont.)

3. Create the directory /etc/grid-security
   3.1 mkdir/etc/grid-security

4. Install the toolkit:
   4.1 ./setuplx.bin
   4.2 For command line, ./setuplx.bin -console
   4.3 Continue through the prompts, select all services to be installed

5. Configure environment:
   3.1 In either /etc/profile (recommended) or ~/.bash_profile
   source /opt/IBMGrid/igt-setenv.sh
   3.2 (Optional) in /etc/bashrc or ~/.bashrc
   alias igt='cd $GLOBUS_LOCATION'

© 2004 IBM Corporation

IGT Installation Steps (cont.)

6. Configure security using openssl

6.1 Login after step 5 or source $GLOBUS_LOCATION/igt-setenv.sh

6.1a May need to copy /usr/share/ssl/openssl.cnf /opt/IBMgrid

6.2 cd /root

6.3 CA.sh -newca

6.4 Hit enter to create a new CA key pair

6.5 Enter the passphrase for the CA - don't forget because this is used to sign certificates

6.6 Need to enter CA information, probably good to determine first

(examples below):

CT=USA

ST=California

O=FNMOC

OU=AMD

CN=FNMOCCA

mail=someonesmail@somewhere.com

6.7 directory demoCA will be created in /root. This is the CA infrastructure, don't lose it or fiddle with the contents, and back up occasionally.

© 2004 IBM Corporation

IGT Installation Steps (cont.)

7. Import the CA into the IGT:

   7.1 igt-import-ca -type PEM -cert /root/demoCA/cacert.pem

8. Request the host cert:

   8.1 grid-cert-request -host <hostname>

9. Sign the host certificate (note if someone feels ambitious these steps can be automated in a script):

   9.1 In the root directory,

   cp /etc/grid-security/hostcert_request.pem ./newreq.pem

   9.2 export SSLEAY_CONFIG='-config /etc/grid-security/globus-user-ssl.conf'

   9.3 CA.sh -sign

   9.4 Follow instructions to sign cert

   9.5 igt-install-certs -host -cert newcert.pem -force

   9.6 Get rid of the .pem files

© 2004 IBM Corporation

IGT Installation Steps (cont.)

10. Request the ibmgrid certificate
    10.1 su - ibmgrid
    10.2 grid-cert-request

11. Sign the ibmgrid certificate
    11.1 As root, cd /root
    11.2 cp ~ibmgrid/.globus/usercert_request.pem ./newreq.pem
    11.3 CA.sh -sign
    11.4 Follow instructions to sign certificate
    11.5 igt-install-certs -user ibmgrid -cert newcert.pem -force
    11.6 Get rid of the .pem files

© 2004 IBM Corporation

IGT Installation Steps (cont.)

12. Test the install:
    - **12.1 Completely log out and log back in as ibmgrid**
    - **12.2 grid-proxy-init -verify -debug**
    - **12.3 igt-start-container**
    - **12.4 After the container is up, check the GSM:**

      http://<host>:12080/gsm

    - **12.5 Test the MMJFS:**

      managed-job-globusrun -factory http://<host>:12080 -file $GLOBUS_LOCATION/etc/test.xml

    - **12.6 If all goes well you should have a stderr and stdout files in ~ibmgrid. The stdout should have some output in it.**

13. To add additional users, follow steps 10-11, and configure grim:
    - **13.1 Edit /etc/grid-security/grim-port-type.xml**
    - **13.2 Add an entry for the new user**

© 2004 IBM Corporation

IGT Installation Steps (cont.)

- CA Server:
  - **To generate other host / user keys follow the example below:**

CA Server	**Rtpmeta.epa.gov**
Castor/cluster	**Castor.nesc.epa.gov**

© 2004 IBM Corporation

IGT Installation Steps (cont.)

- **Rtpmeta> (as root) scp /root/demoCA/cacert.pem ibmgrid@castor.nesc.epa.gov:/tmp/cacert.pem**

- **Castor> (as root) igt-import-ca -type PEM -cert /tmp/cacert.pem**
- **Castor> (as root) rm /tmp/*.pem**
- **Castor> (as root) grid-cert-request -host castor.nesc.epa.gov**
- **Castor> (as root) scp /etc/grid-security/hostcert_request.pem ibmgrid@rtpmeta.epa.gov:/tmp/castor_newreq.pem**
- **Castor> (as root) rm /tmp/*.pem**

- **Rtpmeta> (as root) mv /tmp/castor_newreq.pem /root/newreq.pem**
- **Rtpmeta> (as root) export SSLEAY_CONFIG="-config /etc/grid-security/globus-user-ssl.conf"**
- **Rtpmeta> (as root) CA.sh -sign**
- **Rtpmeta> (as root) scp /root/newcert.pem ibmgrid@castor.nesc.epa.gov:/tmp/castor_newcert.pem**
- **Rtpmeta> (as root) rm /tmp/*.pem**
- **Rtpmeta> (as root) rm /root/*.pem**

© 2004 IBM Corporation

IGT Installation Steps (cont.)

- **Castor> (as root) cd /root (home directory)**
- **Castor> (as root) mv /tmp/castor_newcert.pem ./newcert.pem**
- **Castor> (as root) igt-install-certs -host -cert newcert.pem -force**
- **Castor> (as root) rm /tmp/*.pem**
- **Castor> (as root) rm /root/*.pem**
- **Castor> (as root) su - ibmgrid**
- **Castor> (as ibmgrid) . /opt/IBMGrid/igt-setenv.sh (may not need to do if sourced in /etc/profile or {user} bash)**
- **Castor> (as ibmgrid) grid-cert-request -force -cn {cn-name} {ie. grid2aspen or grid3meta}**
- **Castor> (as ibmgrid) exit**
- **Castor> (as root) scp ~ibmgrid/.globus/usercert_request.pem ibmgrid@rtpmeta.epa.gov:/tmp/ig_newreq.pem**

© 2004 IBM Corporation

IGT Installation Steps (cont.)

- **Rtpmeta> (as root) mv /tmp/ig_newreq.pem /root/newreq.pem**
- **Rtpmeta> (as root) CA.sh -sign**
- **Rtpmeta> (as root) scp /root/newcert.pem ibmgrid@castor.nesc.epa.gov:/tmp/ig_newcert.pem**

- **Rtpmeta> (as root) rm /tmp/*.pem**
- **Rtpmeta> (as root) rm /root/*.pem**

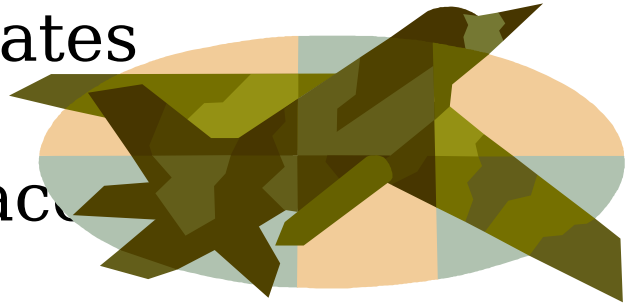
- **Castor> (as root) mv /tmp/ig_newcert.pem /root/newcert.pem**
- **Castor> (as root) cd /root (home directory)**
- **Castor> (as root) igt-install-certs -user ibmgrid -cert newcert.pem -force**
- **Castor> (as root) rm /tmp/*.pem**
- **Castor> (as root) rm /root/*.pem**

© 2004 IBM Corporation

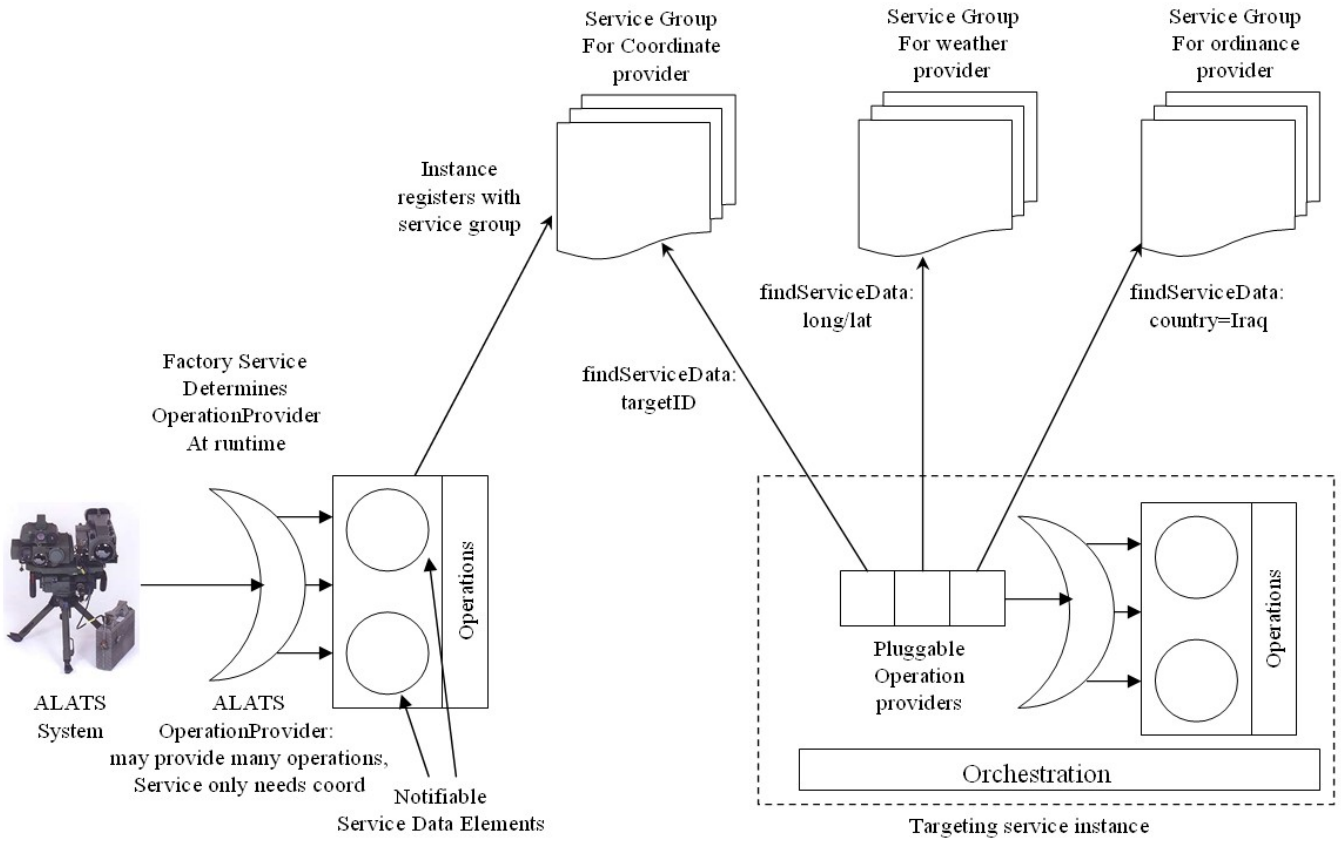
Module 4 – Administration & Developmer

Building a Targeting Service with the IBM Grid Toolbox

- Contribute to battle space activities, situational awareness
- Provides ability to coordinate acquisition, track, and put fire on a target
- Uses other enterprise battlespace services and service groups to identify resources
  - GPS services for coordinates
  - Reconnaissance services
  - Ordinance location and ac
  - Weather services



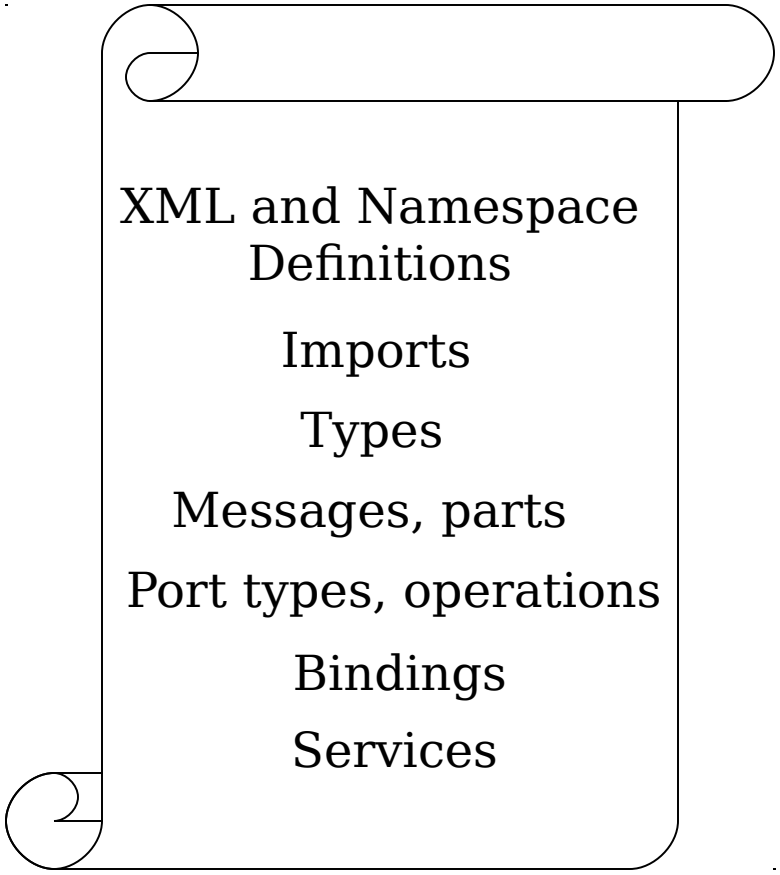
Targeting Service



Its better to directly define the WSDL and GWSDL rather than derive from an interface language (java, idl).

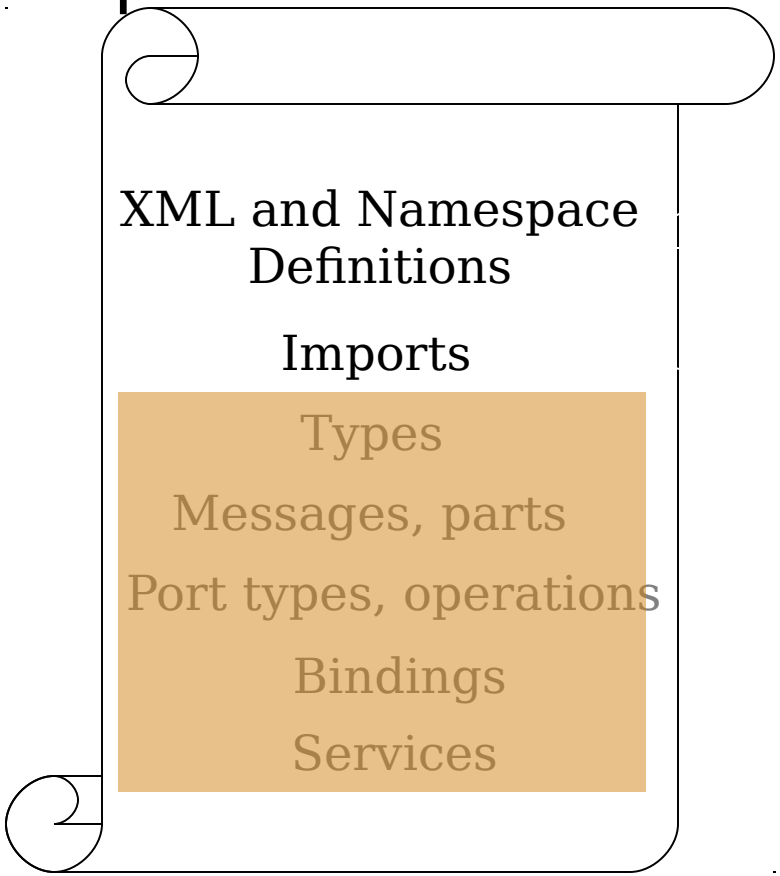
- WSDL is semantically richer
  - More expressive than IDL
  - Broader range of requirements than can be expressed in java
- More control
  - Translation programs (Java2WSDL, etc) limit control you have of final result due to the introduction of an abstraction layer

GWSDL, or Grid WSDL, is an extension of WSDL used in OGSI specification and is used for defining Grid services.



GWSDL document and parts

Declarations specify OGSI namespaces and imports



GWSDL document and parts

Target Namespace:

http://www.targetingc2c.mil/namespaces/0.2/core/gwsdl/Targeting

OGSI Namespaces:

xmlns:ogsi=http://www.gridforum.org/namespaces/2003/03/OGSI
xmlns:gwsdl="http://www.gridforum.org/namespaces/2003/03/gridWSDLExtensions

Import OGSI types, messages, portTy

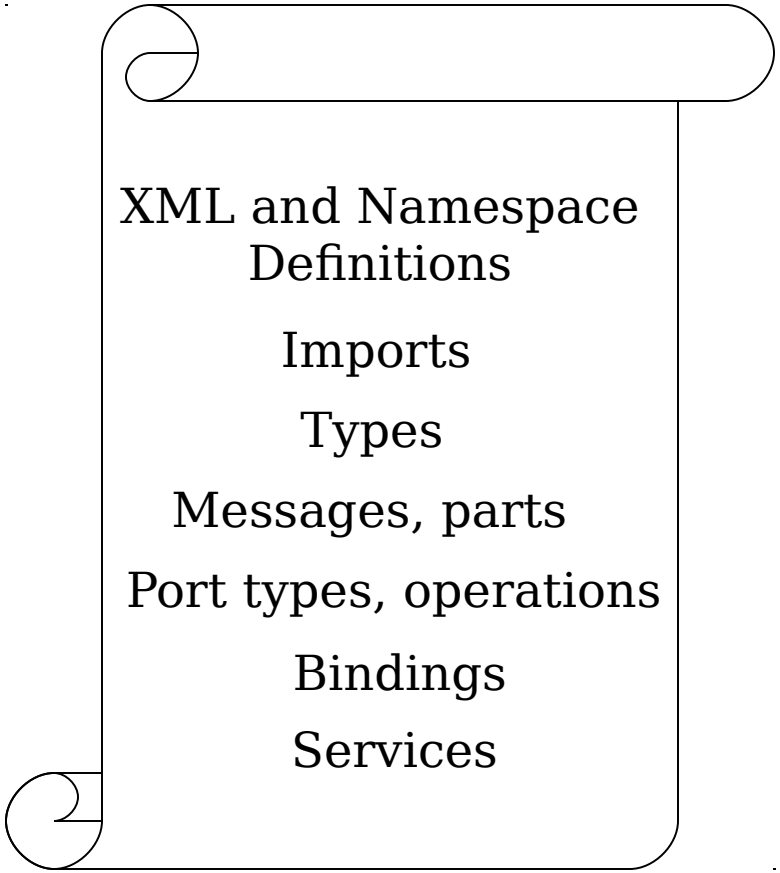
```
<import location="../../ogsi/ogsi.gwsdl"
namespace="http://www.gridforum.org/namespaces/2003/03/OGSI"/>
```

Complete Header

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="TargetService"
                    targetNamespace="http://www.targetingc2c.mil/namespaces/0.2/core/gwsd
                    xmlns:tns="http://www.targetingc2c.mil/namespaces/0.2/core/gwsdl/Targeti
                    xmlns:ogsi="http://www.gridforum.org/namespaces/2003/03/OGSI"
                    xmlns:gwsdl="http://www.gridforum.org/namespaces/2003/03/gridWSDLEx
                    xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                    xmlns="http://schemas.xmlsoap.org/wsdl/">

<import location="../../ogsi/ogsi.gwsdl"
                    namespace="http://www.gridforum.org/namespaces/2003/03/OGSI"/>
```

Differences between WSDL and GWSDL



- GWSDL is extension of WSDL
  - Provides features not available in WSDL 1.1
  - Port type inheritance
    - All grid services extend GridService and implement the interface
  - Service data
    - Needed for maintaining service state in service instances
  - Temporary solution until WSDL 1.2
- No bindings specified
  - Generated by IBM GTB

Mapping namespaces to packages for stub generation

- During build process, stubs are generated which are used for server and client build and deployment
- Stub generator can be configured to map namespaces to packages using a mapping file: *namespace2package.mappings*

http://www.targetingc2c.mil/namespaces/0.2/core/gwsdl/Targeting=cii.core.gwsdl.wsdl
http://www. targetingc2c.mil /namespaces/0.2/core/gwsdl/Targeting/bindings=cii.core.gwsdl.wsdl.bindings
http://www. targetingc2c.mil /namespaces/0.2/core/gwsdl/Targeting/service=cii.core.gwsdl.wsdl.service

- Place in root of dev environment

There are 2 service implementation approaches: inheritance and delegation.

- Inheritance
  - Grid services extend from GridServiceImpl
    - Provides basic grid service functionality
- Delegation
  - Allows operations of port type to be distributed among several classes
  - Each class is called an *operation provider*
  - Operation providers do not extend GridServiceImpl
    - Configure grid service container to use GridServiceImpl to supply basic functionality – done in deployment descriptor
  - Can extend from another class, instead of GridServiceImpl
  - More modular, reusable, and uncoupled designs

Delegation with Operation Providers vs Inheritance

Implementation of OperationProvider interface in a service class

```
import org.globus.ogsa.GridServiceBase;
import org.globus.ogsa.GridServiceException;
import org.globus.ogsa.OperationProvider;
import org.globus.ogsa.GridContext;

import java.rmi.RemoteException;
import javax.xml.namespace.QName;

public class TargetingImpl implements OperationProvider {

  // Operation provider properties
  private static final QName[] operations = new QName[]
{new QName("", "*")};
  private GridServiceBase base;

  // Operation Provider methods
  public void initialize(GridServiceBase base) throws
GridServiceException
  {
    this.base = base;
  }

  public QName[] getOperations()
  {
                return operations;
  }
```

Delegation with Operation Providers vs Inheritance

Deployment descriptor for OperationProvider

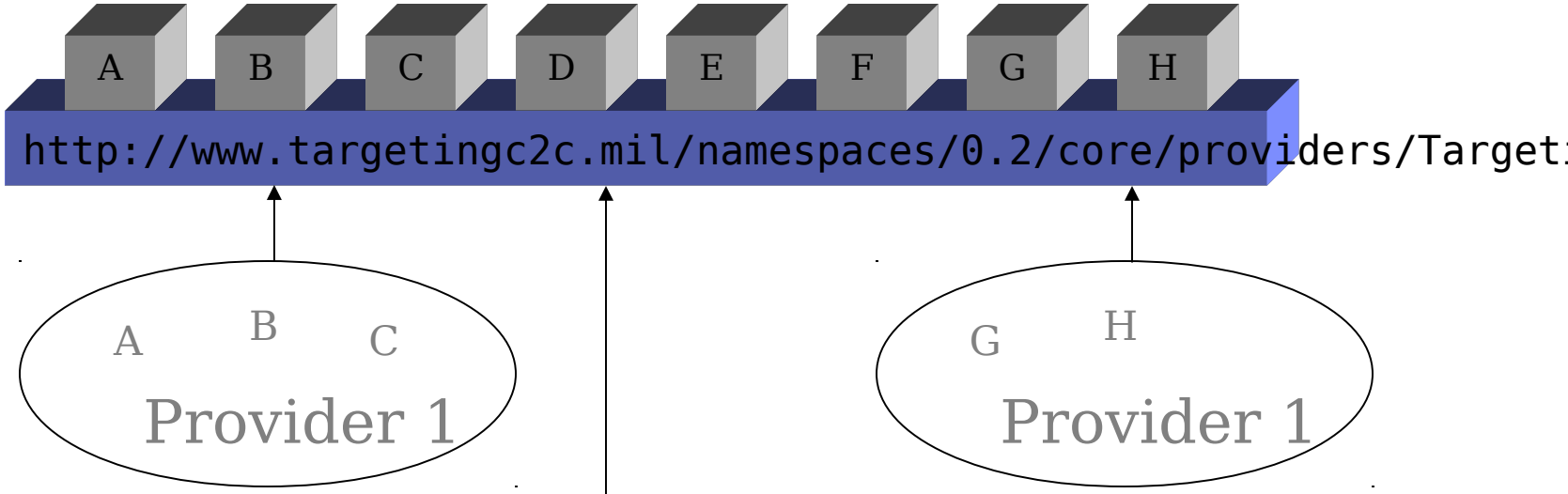
```
<?xml version="1.0"?>
<deployment name="defaultSetverConfig" xmlns="http://xml.apache.org/axis/wsdd/"
                xmlns:java="http://xml.apache.org/axis/wsdd/providers/java">

                <service name="cii/core/providers/TargetingFactoryService" provider="Handler" style="wrapped">
                        <parameter name="name"
                                        value="TargetingService Factory"/>
                        <parameter name="instance-name"
                                        value="TargetingService Instance"/>
                        <parameter name="instance-schemaPath"
                                        value="schema/cii.core.providers/Targeting/Targeting_service.wsdl"/>
                        <parameter name="instance-className"
                                        value="cii.core.providers.wsdl.TargetingPortType"/>
                        <parameter name="instance-operationProviders"
                                        value="cii.core.providers.impl.TargetingProvider"/>
                        <parameter name="instance-baseClassName"
                                        value="org.globus.ogsa.impl.ogsi.GridServiceImpl"/>

                        <!-- Start common parameters -->
                        <parameter name="allowedMethods"
                                        value="*"/>
                        <parameter name="persistent"
                                        value="true"/>
                        <parameter name="className"
                                        value="org.gridforum.ogsi.Factory"/>
                        <parameter name="baseClassName"
                                        value="org.globus.ogsa.impl.ogsi.PersistentGridServiceImpl"/>
                        <parameter name="schemaPath"
                                        value="schema/ogsi/ogsi_factory_service.wsdl"/>
                        <parameter name="handlerClass"
                                        value="org.globus.ogsa.handlers.RPCURIProvider"/>
                        <parameter name="factoryCallback"
                                        value="org.globus.ogsa.impl.ogsi.DynamicFactoryCallbackImpl"/>
                        <parameter name="operationProviders"
                                        value="org.globus.ogsa.impl.ogsi.FactoryProvider"/>
                </service>

</deployment>
```

Delegation and Operation Providers



Binding at deploy time or runtime

Can add providers at runtime:

GridServiceBase.addOperationProviders(OperationProvider)

Implements OperationProvider interface

- Public Qname getOperations()
- Public void initialize(GridServiceBase)
  - GridServiceBase provides grid functionality that inheritance pattern provided by extending GridServiceImpl

If using service data, must delegate method findServiceData(ExtensibilityType) to GridServiceBase.

Module 5 – Extensions

What we will be talking about in this Module

- A look at the IBM Grid Toolbox basic functionality
- What extra functionality is needed for complex grid infrastructures?
- Job scheduling using fork() MMJFS
- Job scheduling using condor MMJFS
- Condor and Condor-G Job Scheduling as a GT3 grid service
- How to create grid services within IGT
- Other technologies which are needed to implement a true production grid infrastructure

© 2004 IBM Corporation

What we will be talking about in this Module

- A look at the IBM Grid Toolbox basic functionality
- What extra functionality is needed for complex grid infrastructures?
- Job scheduling using fork() MMJFS
- Job scheduling using condor MMJFS
- Condor and Condor-G Job Scheduling as a GT3 grid service
- How to create grid services within IGT
- Other technologies which are needed to implement a true production grid infrastructure

© 2004 IBM Corporation

A Quick Recap of the IBM Grid Toolbox

- The IBM Grid toolbox was written to provide a quick start implementation of the Globus Toolkit
- The Globus Toolkit is just that… a toolkit!
- IGT is a self contained installation including
  - A web application server
  - A database
  - A J2EE container
  - ..and some out of the box Grid Services
- ..But..

© 2004 IBM Corporation

IGT in it's 'Out of the Box' form is limited

- The IGT is fairly specific as to what it can do out of the box
- IGT installs the core grid services
- It's an 'all or nothing' approach
- What is missing?
  - Job scheduling
  - Reliable job scheduling, recovery, automatic restarts
  - Charge backs, Billing and Metering
  - Portal Services

- Let's look at the components which make up an average Grid Infrastructure you could expect to see in production…

© 2004 IBM Corporation

What we will be talking about in this Module

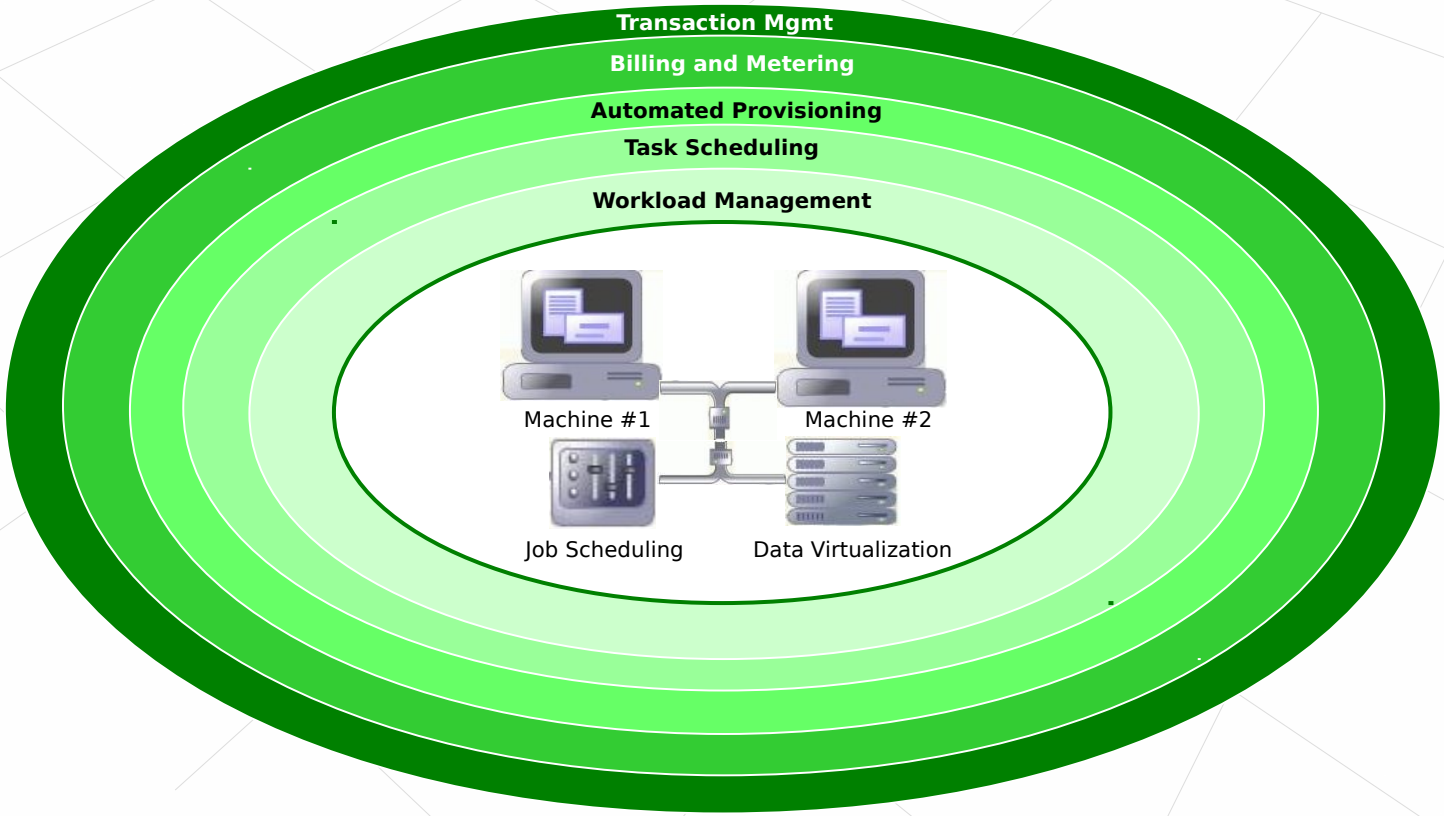
- A look at the IBM Grid Toolbox basic functionality
- What extra functionality is needed for complex grid infrastructures?
- Job scheduling using fork() MMJFS
- Job scheduling using condor MMJFS
- Condor and Condor-G Job Scheduling as a GT3 grid service
- How to create grid services within IGT
- Other technologies which are needed to implement a true production grid infrastructure

© 2004 IBM Corporation

Grid Adoption Steps – Setting the bar

Transaction Management:

- **Manage the execution of e-business transactions across distributed resources**
- **Enable dynamic allocation of resources for transactional and parallel application models**



© 2004 IBM Corporation

Let us concentrate Job Scheduling

- The majority of the Grid ISVs have ‘GT3 compliant’ grid services for job scheduling
  - Platform LSF
  - Condor
  - Maui
  - PBS
- The IGT can be used as the base platform to install these grid services to enable a ‘hook’ into these schedulers.
- Remember the MMJFS – IGT comes with a ‘fork()’ scheduler only.
- Addition of other job schedulers involve creating new MMJFSs

© 2004 IBM Corporation

What we will be talking about in this Module

- A look at the IBM Grid Toolbox basic functionality
- What extra functionality is needed for complex grid infrastructures?
- Job scheduling using fork() MMJFS
- Job scheduling using condor MMJFS
- Condor and Condor-G Job Scheduling as a GT3 grid service
- How to create grid services within IGT
- Other technologies which are needed to implement a true production grid infrastructure

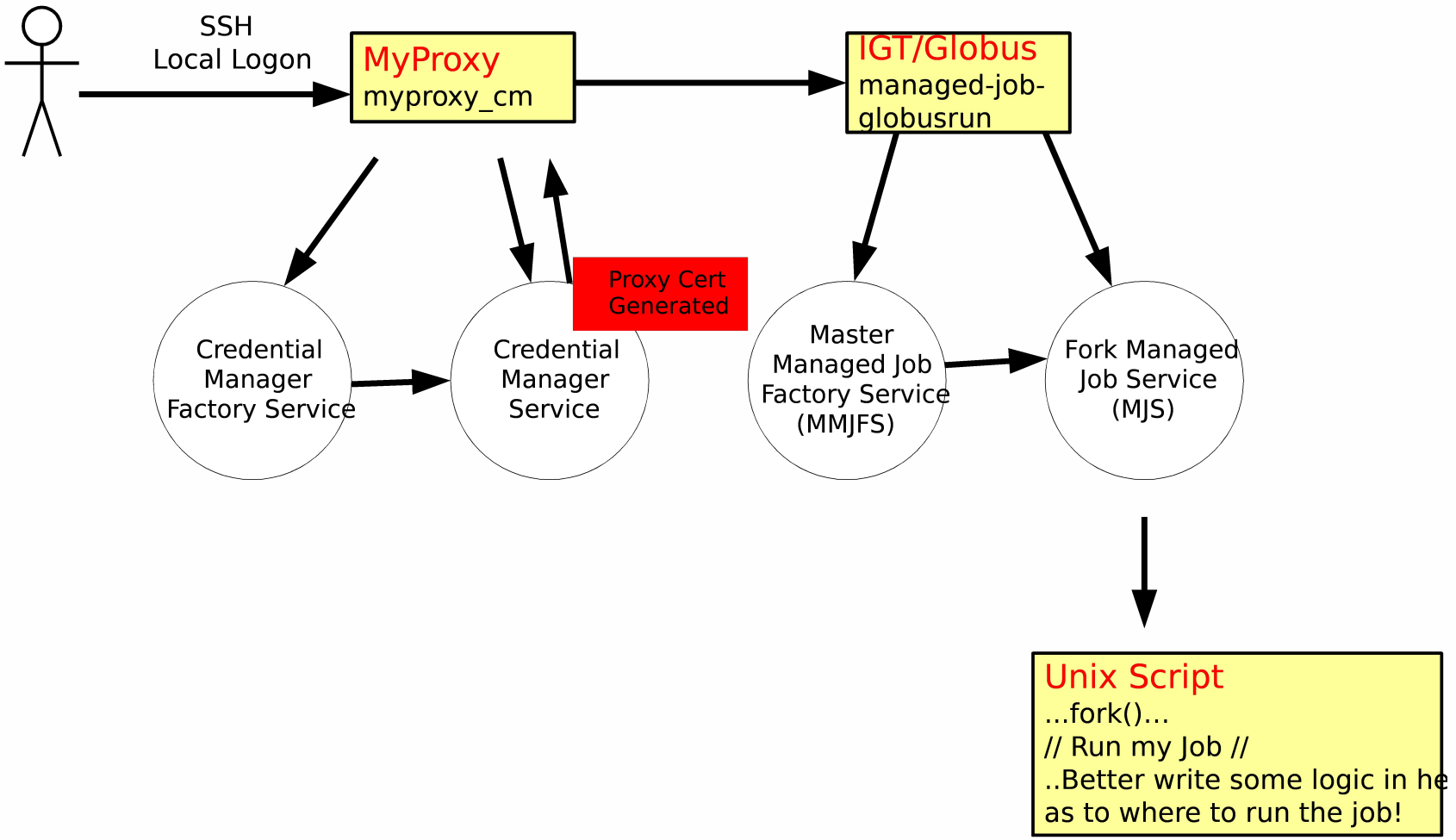
© 2004 IBM Corporation

How does job scheduling work with base IGT?

- It uses the fork() MMJFS
- We have to code scheduling algorithm as a script including
  - Statistics generation
  - Retries
  - failover
- We’re out of the ‘GSI’ domain
- How do you query the grid service?
- How do we stop/restart/suspend the job?

© 2004 IBM Corporation

Job Scheduling using IGT with fork() MMJFS



© 2004 IBM Corporation

What we will be talking about in this Module

- A look at the IBM Grid Toolbox basic functionality
- What extra functionality is needed for complex grid infrastructures?
- Job scheduling using fork() MMJFS
- Job scheduling using condor MMJFS
- Condor and Condor-G Job Scheduling as a GT3 grid service
- How to create grid services within IGT
- Other technologies which are needed to implement a true production grid infrastructure

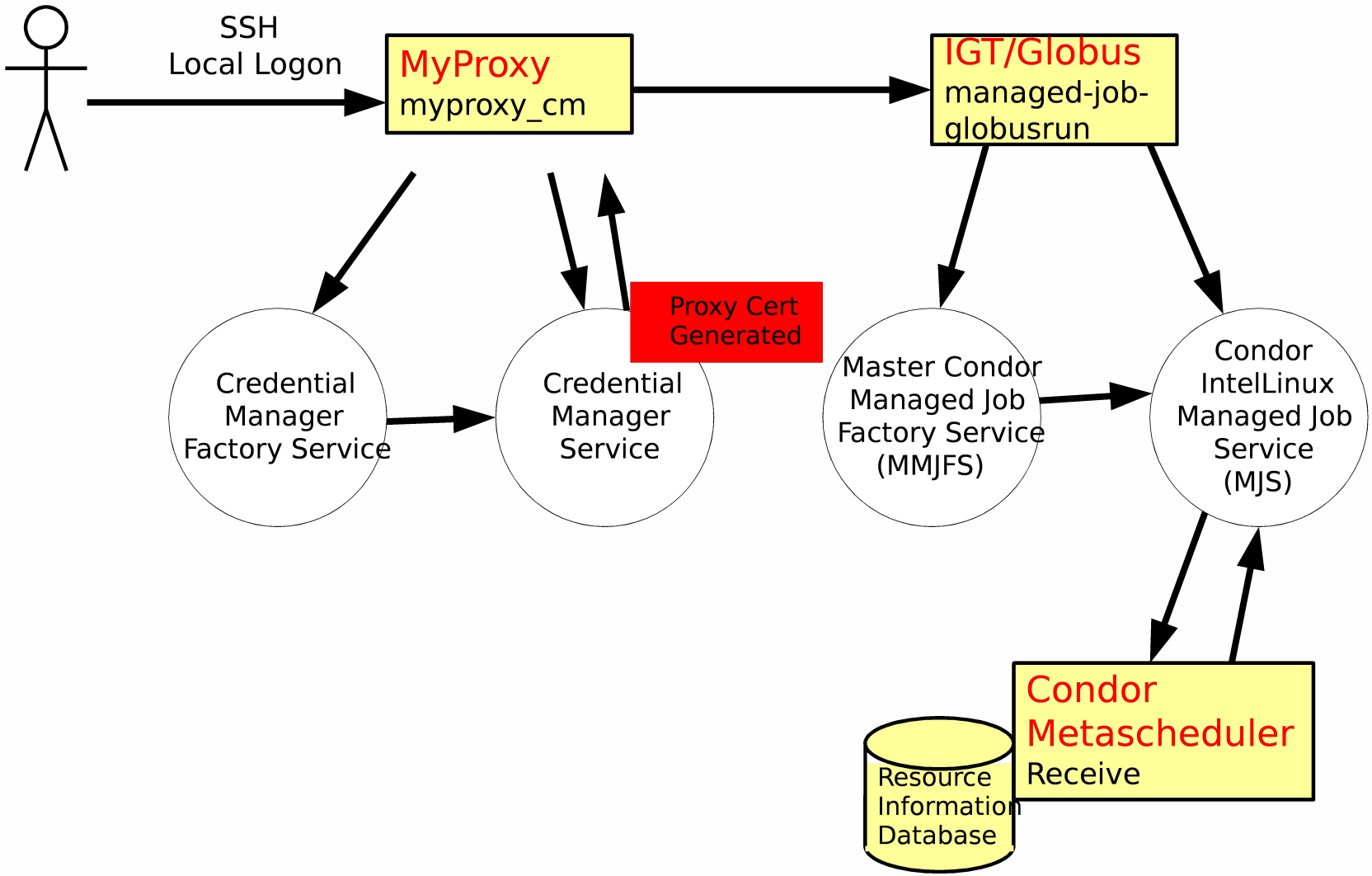
© 2004 IBM Corporation

Job Scheduling using an extension to IGT

- We can install a new MMJFS for Condor,Maui, PBS
- The way that the MMJFS is called is STILL the same open standard. The request for service does not have to care which job scheduler you have picked
- A lot of the work is done for us because the MMJFS is a web service.
- We’re still in of the ‘GSI’ domain
- Standard query the grid service is no problem
- Far more simple to stop/restart/suspend the jobs
- We’re not just fronting a scheduler with a grid service!

© 2004 IBM Corporation

Job Scheduling using IGT with Condor MMJFS



© 2004 IBM Corporation

What we will be talking about in this Module

- A look at the IBM Grid Toolbox basic functionality
- What extra functionality is needed for complex grid infrastructures?
- Job scheduling using fork() MMJFS
- Job scheduling using condor MMJFS
- Condor and Condor-G Job Scheduling as a GT3 grid service
- How to create grid services within IGT
- Other technologies which are needed to implement a true production grid infrastructure

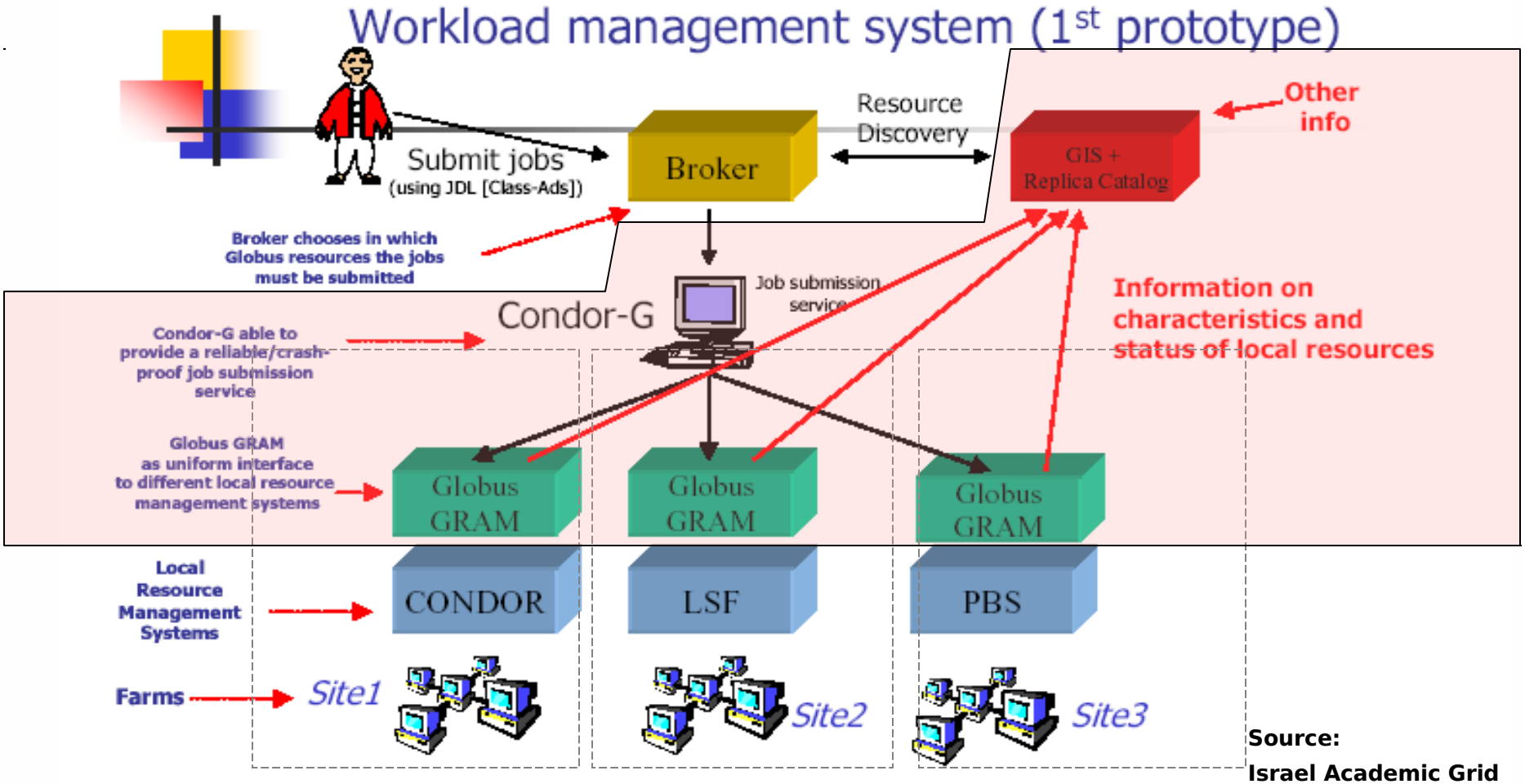
© 2004 IBM Corporation

A look at one GT3 Job Scheduler - Condor

- **Guru:**
  - **Miron Livny**
- **Condor Project:**
  - **http://www.cs.wisc.edu/condor**
- **Condor-G manual:**
  - http://www.cs.wisc.edu/condor/manual/v6.4/5_2Condor_G.html
- **Condor is a scheduler, similar to PBS, LSF and others**
- **Condor-G is (the submission) part of Condor.**
- **It adds to Globus Reliable job submission**

© 2004 IBM Corporation

Condor-G explained



Source - Condor

Source:

Israel Academic Grid (IAG)
Itzhak Ben Akiva (TAU)

David Front (WI)

© 2004 IBM Corporation

What we will be talking about in this Module

- A look at the IBM Grid Toolbox basic functionality
- What extra functionality is needed for complex grid infrastructures?
- Job scheduling using fork() MMJFS
- Job scheduling using condor MMJFS
- Condor and Condor-G Job Scheduling as a GT3 grid service
- How to create grid services within IGT
- Other technologies which are needed to implement a true production grid infrastructure

© 2004 IBM Corporation

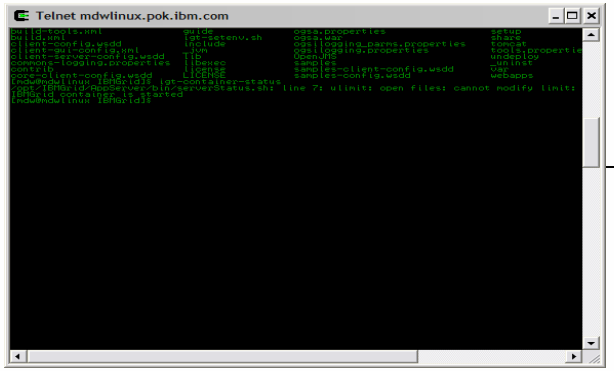
The IGT allows easy creation of new Services

- **Create a grid service from existing Java code. A single command, creates all of the artifacts needed to deploy the new service**
- **Package services and related artifacts into a Grid Service Archive (“GAR”) for deployment into the runtime environment.**
- **Easily integrated into an Apache Ant build environment.**

© 2004 IBM Corporation

Grid Service Development using Dev Tools

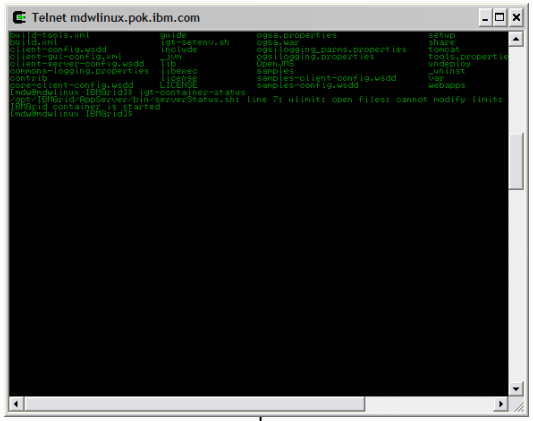
1 $ ant createService

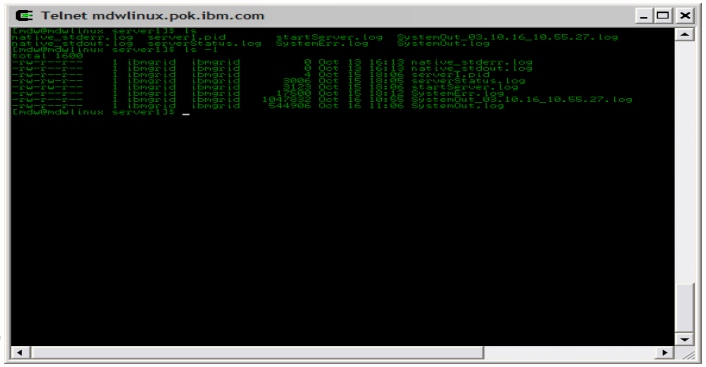


WSDL
Jars / EJB Jars
Deployment Desc

Grid Service Archive
(gar)

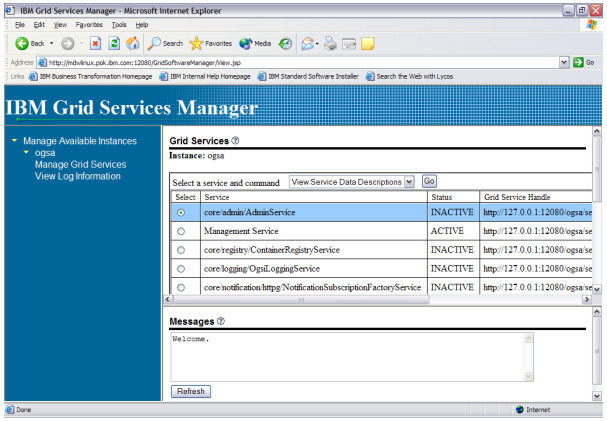
2 $ igt-deploy-gar





3 $ igt-start-container

4 View and Interact with the service using the Grid Services Manager



© 2004 IBM Corporation

How do we call the new MMJFS?

- Managed-globus-jobrun
- To managed-globus-jobrun – factory <new>
- RSL file is changed to allow condor job scheduling

© 2004 IBM Corporation

What we will be talking about in this Module

- A look at the IBM Grid Toolbox basic functionality
- What extra functionality is needed for complex grid infrastructures?
- Job scheduling using fork() MMJFS
- Job scheduling using condor MMJFS
- Condor and Condor-G Job Scheduling as a GT3 grid service
- How to create grid services within IGT
- Other technologies which are needed to implement a true production grid infrastructure

© 2004 IBM Corporation

Of Course, it's not just about Grid Services

- A production grid infrastructure can utilize multiple technologies – not just those which have to be grid services
- Data Grids need to implement Avaki, GPFS, NFS etc
- Local cluster management using Condor, LoadLeveler, etc
- Portal Services using WebSphere Portal Server, TomCat, etc…
- Systems Management, Monitoring, Change Mgmt

- … It's what you want to expose to 3rd parties or users which need to be open standards based grid services.

- A production grid infrastructure will most likely be made of up multiple components, IGT could be just one

© 2004 IBM Corporation

Module 6 – Customer Examples

© 2004 IBM Corporation

Introduction to EPA

- A Compute and Data Grid for the Environmental Protection Agency (EPA)
- Agency provides applications and supporting infrastructure to researchers worldwide
- Researchers are able to perform complex environmental computations on agencies infrastructure – the results of which are used by different states
- CMAQ application was chosen as the first for grid enablement
  - CMAQ was already running on two clusters
- EPA was already familiar with condor as a cluster scheduler. Architectural decision was made to use condor as meta scheduler in grid infrastructure

© 2004 IBM Corporation

Why did EPA want to implement Grid?

- Agency had the following pain points
  - Applications are silod
  - Utilization rates of applications are low
  - No easy integration between researchers and applications
  - Multiple access to same information
- Wanted to adopt a services orientated architecture – ‘pay by the drink’
- Wanted a effective method to allow better collaboration between researchers
- EPA wanted both a compute and data centric grid infrastructure pilot to prove the value of grid computing
- EPA wanted to conform to open standards
- EPA wanted to use current infrastructure with minimal change
- First application to be chosen was called ‘CMAQ’
- IBM and CSC worked with IBM to produce a working prototype

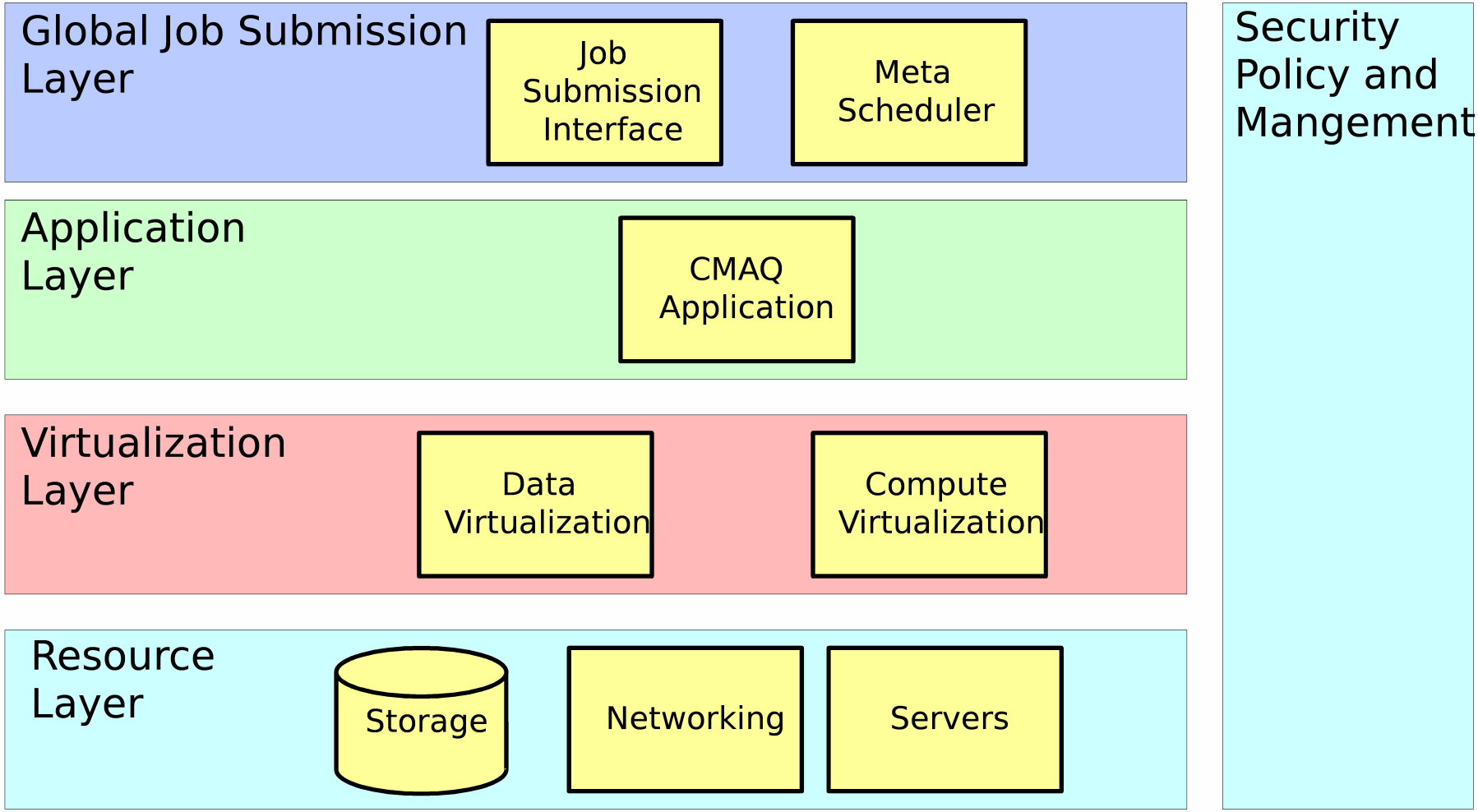
© 2004 IBM Corporation

Why was the IGT chosen?

- Allows a services orientated architecture
- Web services enabled infrastructure adheres to open standards
- Extensible
- Fast to deploy
- Open Source
- Allowed them to create the grid services needed
- They wanted to be able to swap in and out different job schedulers – and not to have to dictate which schedulers clustered applications should use

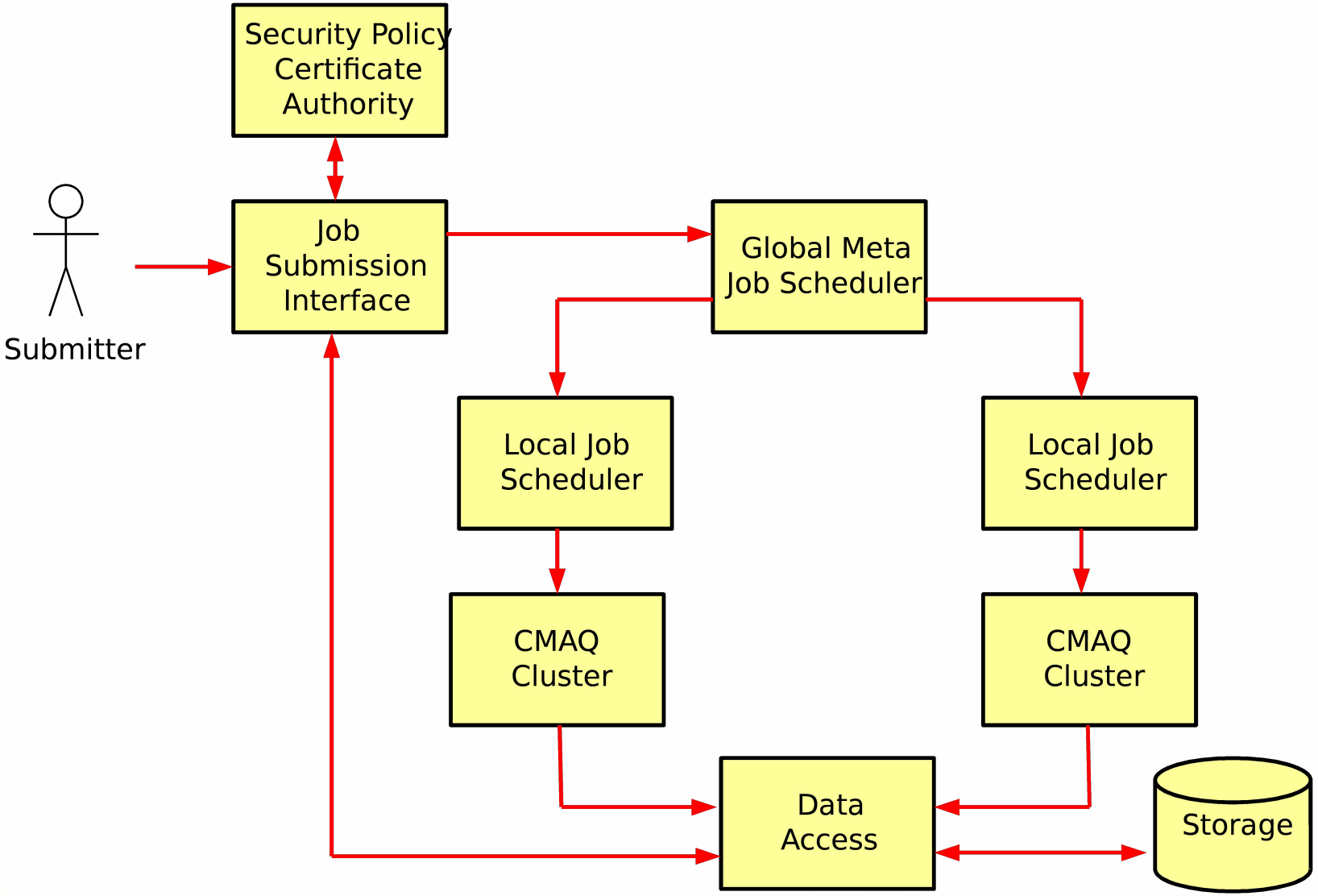
© 2004 IBM Corporation

Layered Approach for EPA



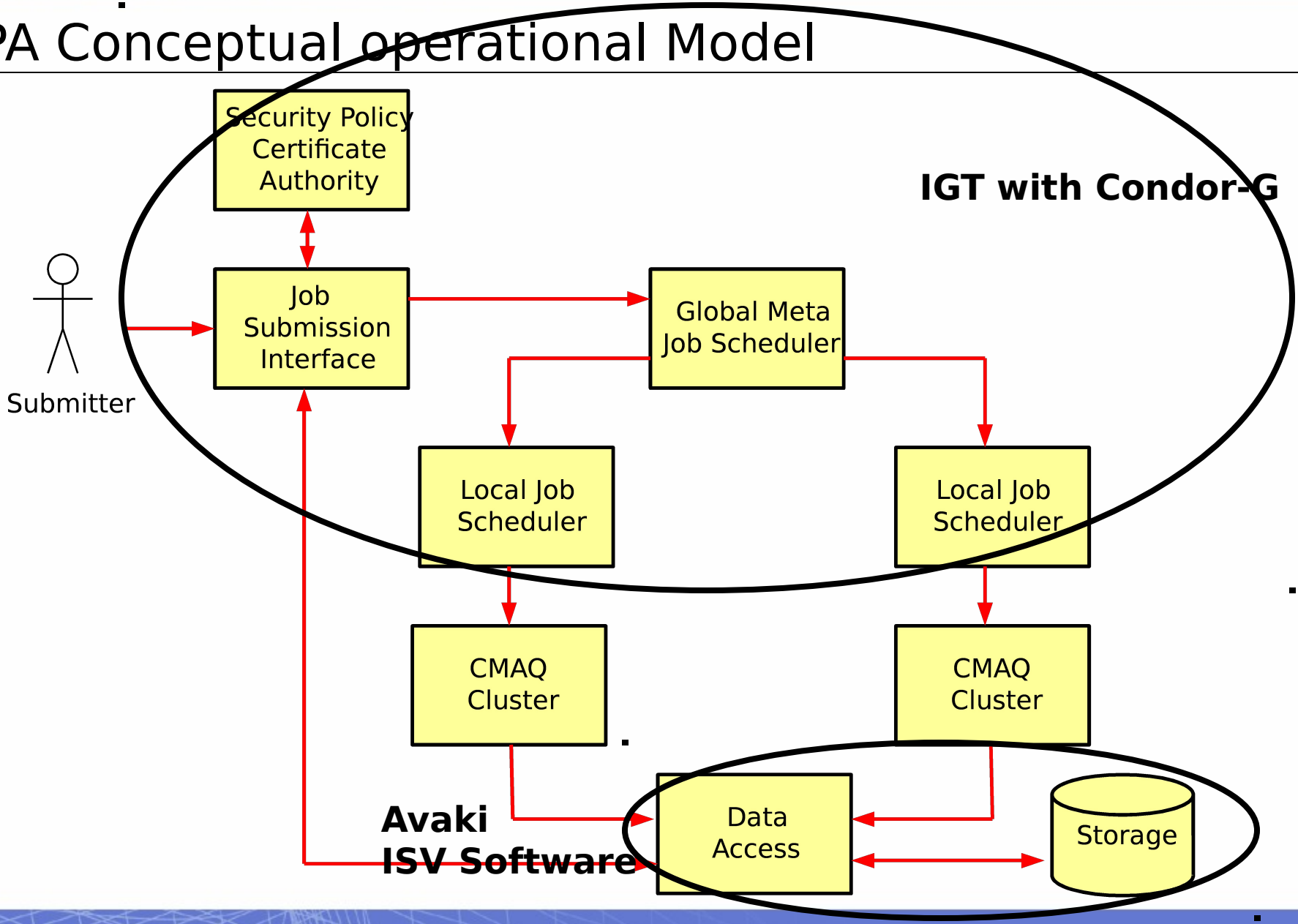
© 2004 IBM Corporation

EPA Conceptual operational Model



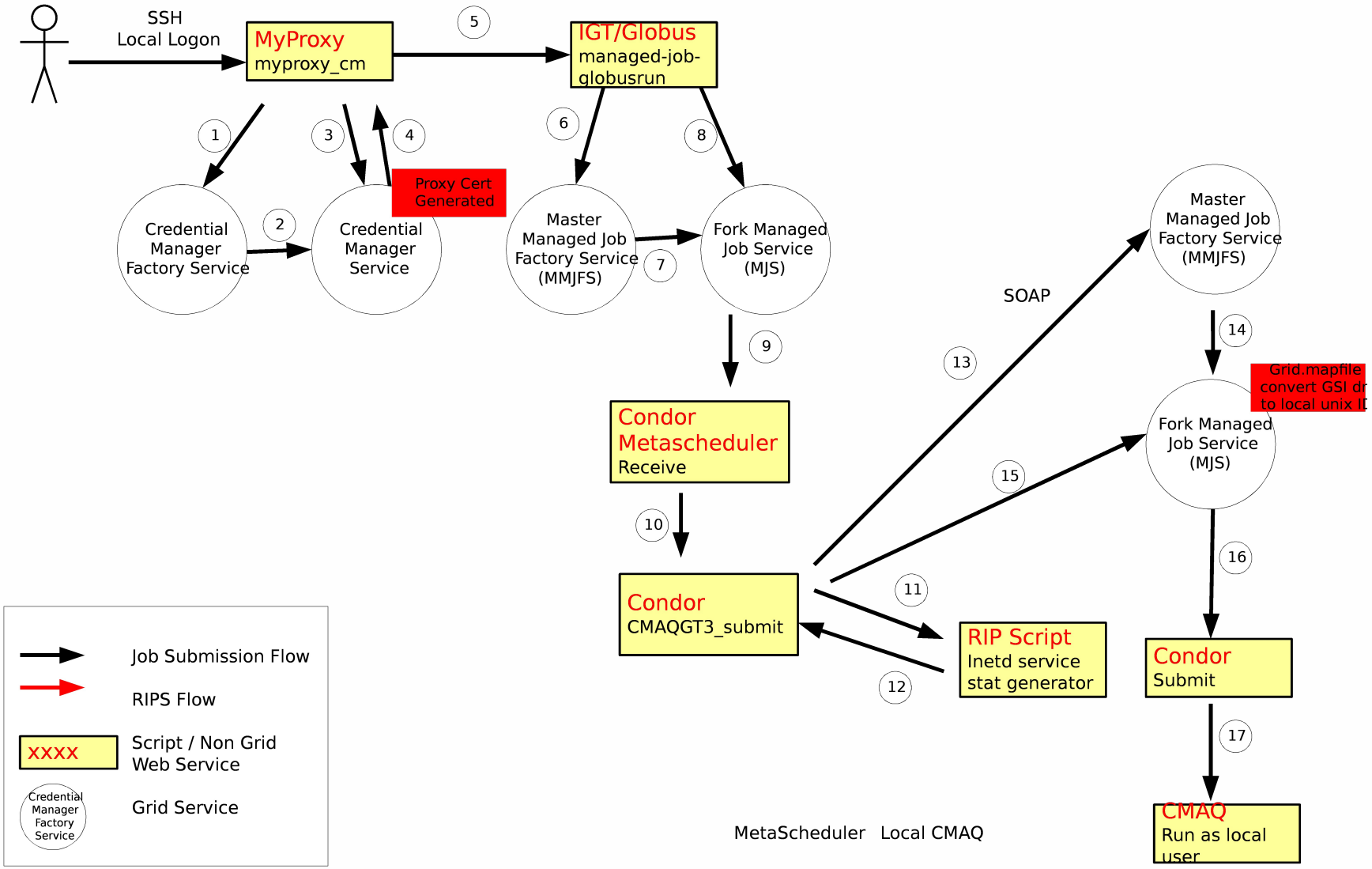
© 2004 IBM Corporation

EPA Conceptual operational Model



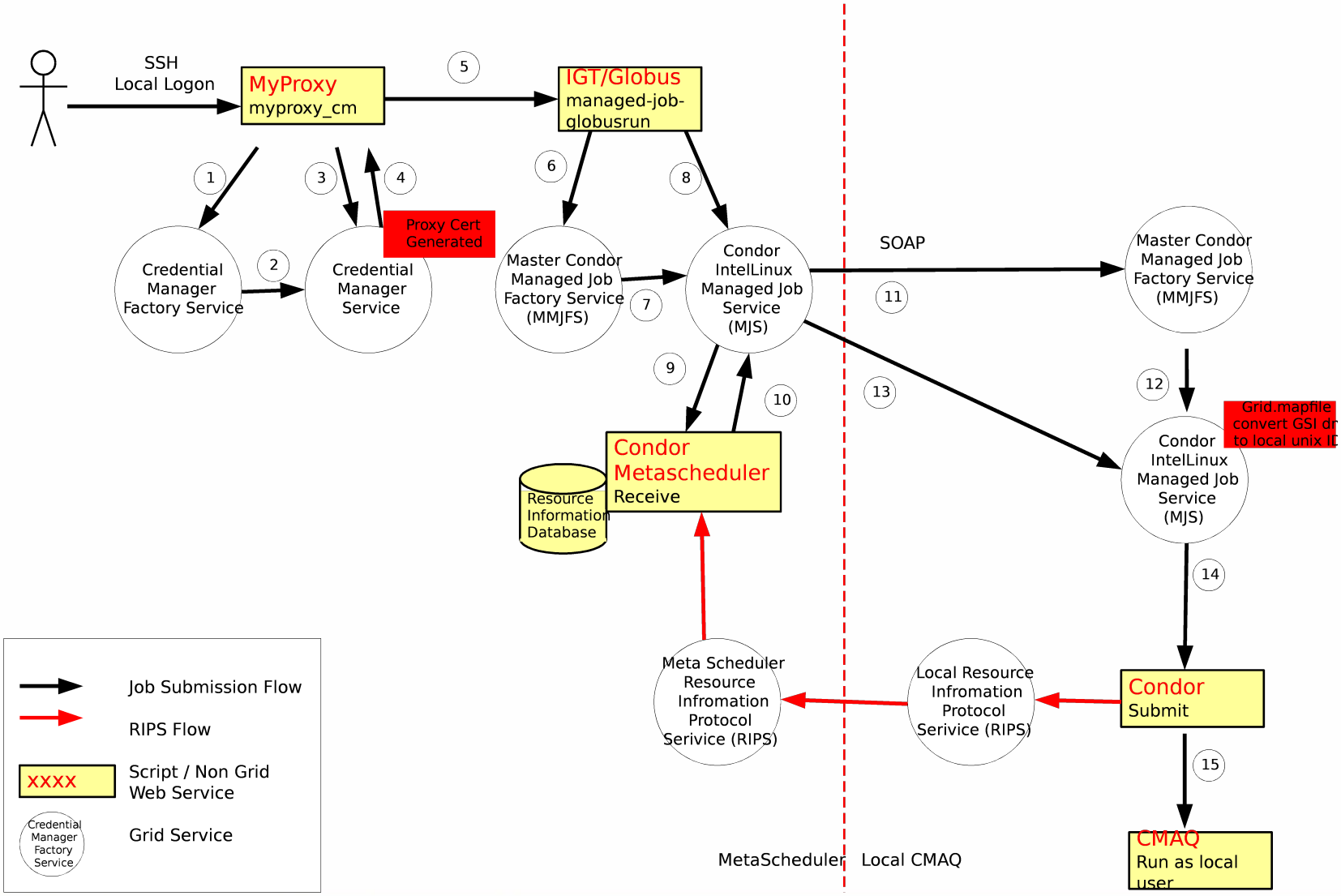
© 2004 IBM Corporation

Job submission with MMJFS Fork()



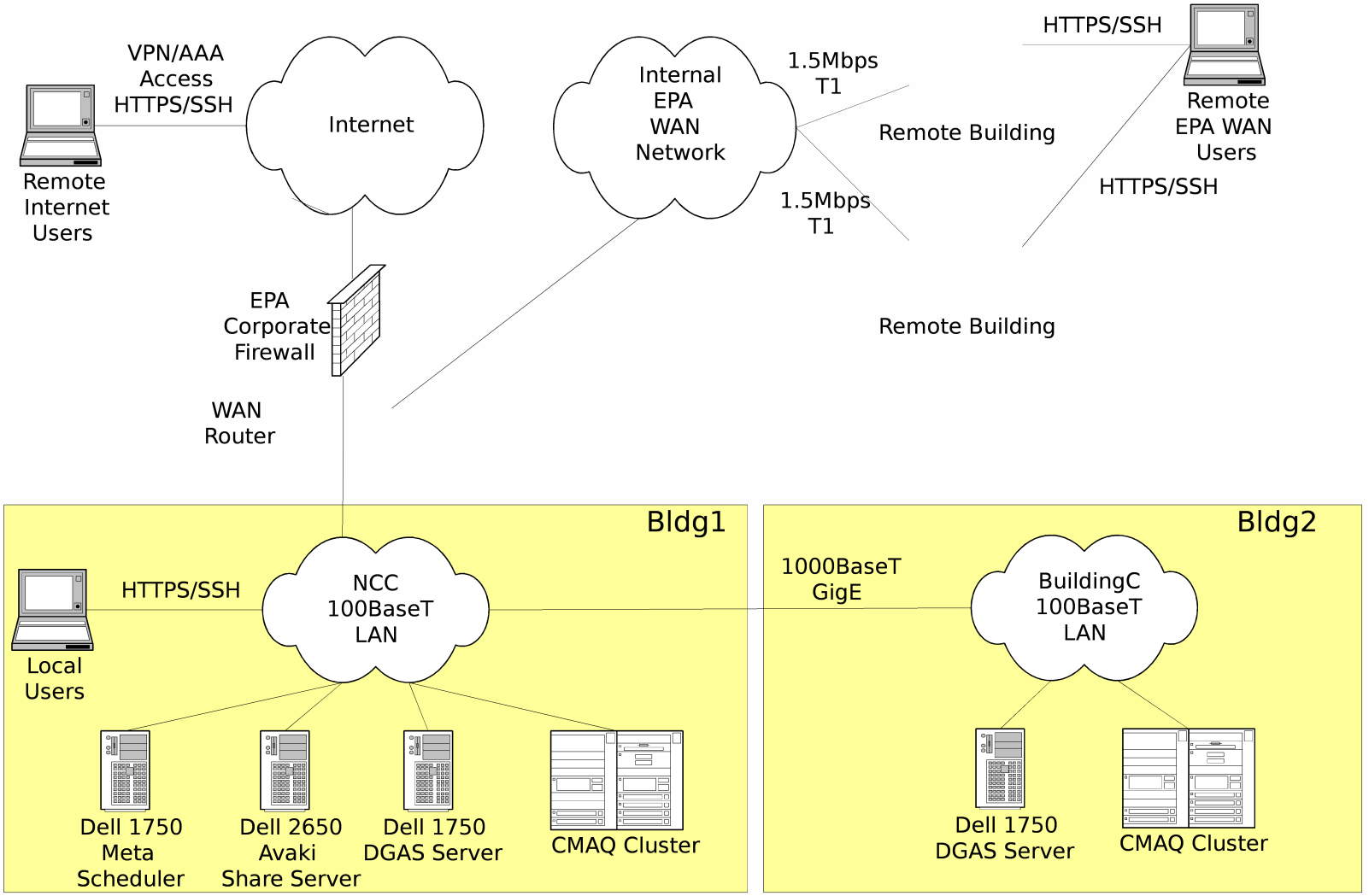
© 2004 IBM Corporation

Job Submission with Condor MMJFS



© 2004 IBM Corporation

The EPA Specified Network Topology



© 2004 IBM
Corporation

EPA can benefit from IGT going forward

- Scalability to more than one application
- Grid technologies allows higher utilization of current resources
- Ease of integration of new applications
- Scheduling and availability already taken into account
- RIPS can be updated for new resource types
- Grid services allow jobs to flow over WAN
- Virtual Organizations using X5.09 for collaboration groups
- Avaki datagrid can expand data centric grid for enhanced data directory

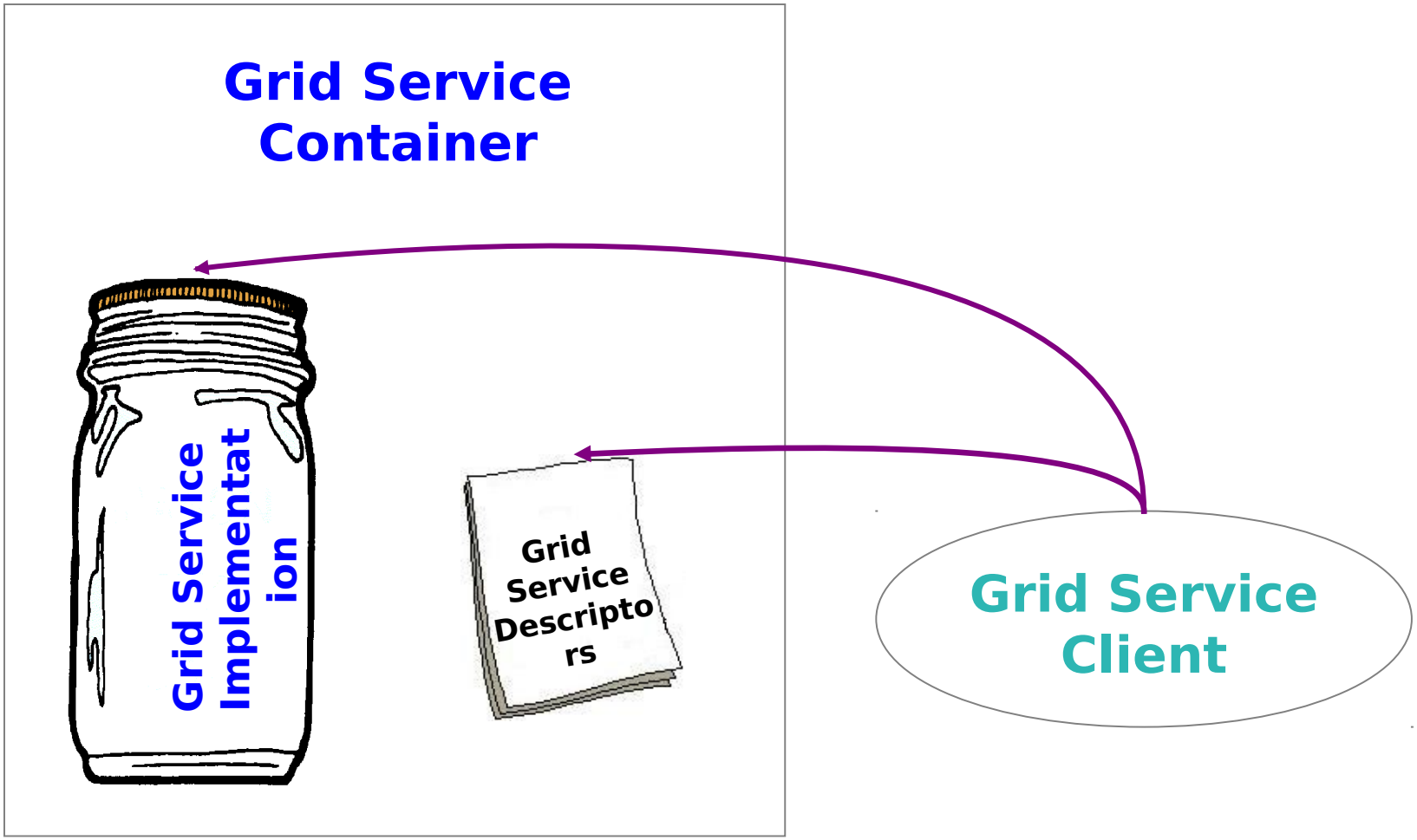
© 2004 IBM Corporation

Homework Assignment

Introduction

- The Open Grid Services Infrastructure (OGSI) specification defines interfaces, semantics and constraints for a Web Service to be considered a Grid Service.
- Grid Services are complex
  - They contain several different components (described in the following slides).
  - In order to author a grid service, a developer starting from scratch would need:
    - extensive knowledge of the OGSI specification
    - details about the grid hosting environment and packaging structure.

Grid Services



What Makes Up a Grid Service? (1 of 3)

- Business Logic
  - The function to be exposed as a grid service
- WSDL Description
  - XML markup that describes:
    - The grid service interface
    - The means by which communication with the grid service occurs (bindings)
    - The location of the grid service (port)
- Operation Providers
  - Classes that enable the function containing the business logic to conform with the OGSI specifications
  - Operation provider methods are wrappers for the methods in the business logic.
    - Allow separation between the grid service and business logic semantics

What Makes Up a Grid Service? (2 of 3)

- Binding Stubs
  - Binding stubs are JAX-RPC compliant interfaces that are used on both the client and server.
  - The stubs allow the client to communicate to the server using specific messages over specific protocols, e.g. SOAP over HTTP.
- Service Locators
  - Locators are used by the client in order to be able to find and use the grid service.
  - Given a service endpoint, the locators return an instance of the Binding stub to the client, which is used to communicate with the grid service.

What Makes Up a Grid Service? (3 of 3)

- Port Type Interface
  - An interface described in code that defines all public methods that make up the grid service.
- Deployment Information
  - WSDD – Web Service Deployment Descriptor
  - A WSDD fragment describes the service, including:
    - The service name
    - The location of the WSDL associated with the service
    - The location of the code associated with the service
  - Upon deployment, this fragment is merged with a master WSDD file that describes the services that are currently deployed.

The Complexity of Grid Service Development

- There are many technologies involved in creating all of these grid service components.
  - WSDL
  - JAX-RPC
  - SOAP/HTTP
  - OGSI Specifications
  - Hosting environment implementation
  - Packaging deployment component
- Tooling
  - Hides the complexities of grid services, allowing the user to focus on the business logic.
  - Provides a complete grid application development environment
  - Allows the developer to focus on business logic and have iterative development

The Grid Service Tooling Solution

- Grid Service Tooling was contributed by IBM to the Globus Alliance and integrated into Globus Toolkit 3.0.
- Grid Service Tooling is available in IBM Grid Toolbox
  - Allows creation of grid service components with less effort and in less time
  - Provides programmatic generation for all grid service components.
  - Non-intrusive to the user's source code.
  - Requires minimal knowledge of OGSI specification.

Grid Service Tooling Features (1 of 2)

- Grid Service Tooling automates the generation of grid service components with a one-step process.
- Allows service creation from two types of sources:
  1. An existing Java class (Bottom-Up).
     - This is the business logic component.
     - The remaining components are generated programmatically.
  2. An existing GWSDL interface (Top-Down).
     - GWSDL is an extension to the existing WSDL specification
     - All parts are generated programmatically.
     - Code must be added to the generated implementation class to serve as the business logic.

Grid Service Tooling Features (2 of 2)

- Generation of code to support Notification Source, Notification Sink, Service Group and Service Group Registration port types.
- Batch grid service creation facility
  - Allows for the creation of multiple grid services, described in an XML file, with a single command
- Simplistic service data tooling
  - Hides the underlying Globus implementation of denoting service data within a grid service
- Batch service data creation facility
  - Allows multiple denotations of service data, described in an XML file, with a single command
- Archiving facility to package all grid service components into a deployable unit
- Requirements:
  - Apache Ant v1.5.x (available from http://ant.apache.org)

Grid Service Development

- Information on developing grid services is shipped with IBM Grid Toolbox
- The eServer Information Center contains a set of tutorials and samples.
- All grid service development commands are Ant targets
  - The Grid Service Development Tools Guide in the eServer Information Center contains a listing of all targets and their properties.

Tooling Ant Targets (1 of 7)

- createBottomUpGridService
  - applicable when the developer has a legacy java application that needs to be made available as a grid service.
  - the part of the application being exposed as a grid service must be a Java class that has been compiled and archived into a jar file.
- Some properties that can be set for this target are:
  - outputDir
  - bottomUp.sourceJar
  - bottomUp.classOfPortType

Tooling Ant Targets (2 of 7)

- createTopDownGridService
  - applicable when a developer:
    - gets a GWSDL description for a grid service from an implementation in another environment or standards community
    - has written the GWSDL, and wants to generate the java mapping for this interface.
- Some properties that can be set for this target are:
  - outputDir
  - topDown.gwsdl-uri

Tooling Ant Targets (3 of 7)

- createGridServicesFromFile
  - applicable when a developer would like to create multiple grid services
  - The developer must create an XML file containing a description of these services. See $GLOBUS_LOCATION/guide/xml/CreateGridServices.xml for an example of such a file.
- The createGridServicesFromFile target in $GLOBUS_LOCATION/guide/build-tools-usage.xml shows an example usage of this command.

Tooling Ant Targets (4 of 7)

- addServiceData
  - Can be used after grid service code has been generated by one of the prior 3 commands
  - Data returned by public accessor methods ,i.e. getXXX(), that are a part of the class containing the business logic, are the only data that can be designated as service data after the grid service code has been generated via one of the creation commands.
- Some properties that can be set for this target are:
  - outputDir
  - serviceData.name
  - serviceData.class

Tooling Ant Targets (5 of 7)

- removeServiceData
  - Can be used after service data has been added using addServiceData
  - If the addition of this service data was in error, this operation can provide a means of correcting the error.
- Some properties that can be set for this target are:
  - outputDir
  - serviceData.name
  - serviceData.class

Tooling Ant Targets (6 of 7)

- createServiceDataFromFile
  - applicable when a developer would like to add multiple service data elements to a grid service
  - The developer must create an XML file containing a description of these services. See $GLOBUS_LOCATION/guide/xml/BatchSD.xml for an example of such a file. Some properties that can be set for this target are:
- The createServiceDataFromFile target in $GLOBUS_LOCATION/guide/build-tools-usage.xml shows an example usage of this command.

Tooling Ant Targets (7 of 7)

- gar (Grid Service Archive)
  - A GAR is an archive file that can serve as the delivery mechanism for grid services
  - It consists of
    - wsdl files
    - tooling-generated java parts (I.e. SOAP binding stubs, port type interface, etc)
    - wsdd fragments
  - can contain multiple tooling generated sets of grid service code
  - is a deployable unit into the Bobcat container
- Some properties that can be set for this target are:
  - outputDir
  - jar.name
  - gar.name

Development considerations

- Which Ant target is appropriate?
- Where is the business logic or gwsdl file that describes the interface?
- Which class contains the business logic?
- In which directory should the tooling output be generated?
- Do any additional port types need to be supported (e.g. notificationSource, serviceGroup)?
  - This is determined by the functionality desired of the grid service.
- Does the service need to be persistent?
  - Persistent grid services do not implement the factory port type; non-persistent services are grid service factories.
- What service data needs to be exposed?
- What is the name of the service data?
- What is the name of the method that returns this data?
- To which jar should Tooling generated output be directed?
- What should the name of the deployable unit be?

Starting with a Legacy Application

- A legacy Java application, Counter, must be made available as a grid service.
- The counter provides the operations *add* and *subtract*.
- Counter exists in a single jar file, counter.jar, with a single class inside, org.globus.ogsa.guide.impl.LegacyCounterImpl

Legacy Application to Grid Service (1 of 5)

- Which Ant target is appropriate?
  - createBottomUpGridService
- Where is the business logic?
  - $GLOBUS_LOCATION/guide/lib/counter.jar
- Which class contains the business logic?
  - org.globus.ogsa.guide.impl.LegacyCounterImpl
- In which directory should the tooling output be generated?
  - $GLOBUS_LOCATION/sample1
- Do any additional port types need to be supported (e.g. notificationSource, serviceGroup)?
  - No
- Does the service need to be persistent?
  - No

Legacy Application to Grid Service (2 of 5)

- What service data needs to be exposed?
  - The value of the counter
- What is the name of the service data?
  - currentValue
- What is the name of the method that returns this data?
  - getValue
- To which jar should Tooling generated output be directed?
  - bottomUpCounterService.jar
- What should the name of the deployable unit be?
  - bottomUpCounterService.gar

Legacy Application to Grid Service (3 of 5)

- Execute the commands:
  - cd /opt/IBMGrid
  - source igt-setenv.sh
  - ant -f build-tools.xml createBottomUpGridService \
    -Dverbose=true \
    -DoutputDir=$GLOBUS_LOCATION\sample1 \
    -DbottomUp.sourceJar=$GLOBUS_LOCATION/guide/lib/counter.jar \
    -DbottomUp.classOfPortType=org.globus.ogsa.guide.impl.LegacyCounterImpl
  - ant –f build-tools.xml addServiceData \
    -Dverbose=true \
    -DoutputDir=$GLOBUS_LOCATION\sample1 \
    -DserviceData.class=org.globus.ogsa.guide.impl.LegacyCounterImpl \
    -DserviceData.class=currentValue -DserviceData.method=getValue
  - ant -f build-tools.xml gar \
    -DoutputDir=$GLOBUS_LOCATION\sample1 \
    -Djar.name=bottomUpCounterService.jar \
    -Dgar.name=bottomUpCounterService.gar
- $GLOBUS_LOCATION/sample1/bottomUpCounterService.gar created!

Legacy Application to Grid Service (4 of 5)

- An alternative to typing in all of those commands is to save the information as Ant targets in an XML file, build-counter-service.xml
- A sample target:

```
<target name="createBottomUpGridService">
    <ant antfile="${build.tools}" target="createBottomUpGridService">
      <property name="verbose" value="true"/>
      <property name="outputDir" value="./sample1"/>
      <property name="bottomUp.sourceJar"
value="guide/lib/counter.jar"/>
      <property name="bottomUp.classOfPortType"
value="org.globus.ogsa.guide.impl.LegacyCounterImpl"/>
    </ant>
 </target>
```

- To complete all Tooling steps, issue the command:
  - `ant –f build-counter-service.xml all`
- See $GLOBUS_LOCATION/guide/build-tools-usage.xml

Legacy Application to Grid Service (5 of 5)

- Deploy the gar:
  - igt-deploy-gar $GLOBUS_LOCATION/sample1/bottomUpCounterService.gar
- Start the container
  - igt-start-container
- The Counter Service is up!
- You will be able to see it in the list of services in Grid Services Manager

Starting with a Grid Service Description

- A grid service must be created from a gwsdl description of a counter service
- counter_port_type.gwsdl describes the counter interface

WordPad Document

Grid Service Description to Grid Service (1 of 5)

- Which ant target is appropriate?
  - createTopDownGridService
- Where is the gwsdl file that describes the interface?
  - $GLOBUS_LOCATION/schema/samples/counter/counter_port_type.gwsdl
- In which directory should the tooling output be generated?
  - $GLOBUS_LOCATION/sample2
- Do any additional port types need to be supported (e.g. notificationSource, serviceGroup)?
  - No

Grid Service Description to Grid Service (2 of 5)

- Does the service need to be persistent?
  - No
- What service data needs to be exposed?
  - None
- To which jar should Tooling generated output be directed?
  - topDownCounterService.jar
- What should the name of the delivery unit (gar file) be?
  - topDownCounterService.gar

Grid Service Description to Grid Service (3 of 5)

- Execute the commands:
  - cd /opt/IBMGrid
  - Source igt-setenv.sh
  - ant -f build-tools.xml createTopDownGridService \
    -Dverbose=true \
    -DoutputDir=$GLOBUS_LOCATION\sample2 \
    -DtopDown.gwsdl-uri=
    $GLOBUS_LOCATION/schema/samples/counter/counter_port_type.gwsdl
- Add the business logic to $GLOBUS_LOCATION/sample2/org/globus/samples/Counter/Counter.java. Details on following slide.
  - ant -f build-tools.xml gar \
    -DoutputDir=$GLOBUS_LOCATION\sample2 \
    -Djar.name=topDownCounterService.jar \
    -Dgar.name=topDownCounterService.gar
- $GLOBUS_LOCATION/sample2/topDownCounterService.gar created!

Grid Service Description to Grid Service (4 of 5)

```
---------- Top of File ----------
/**
 * Counter.java
 *
 * This file was auto-generated by grid services tooling for the
 *  Globus Toolkit
 *
 */

package org.globus.ogsa.samples.counter;

public class Counter {
    public Counter() {
    }

    public int getValue() {
        return -3;
    }

    public int add(int value) {
        return -3;
    }

    public int subtract(int value) {
        return -3;
    }

}
```

```
---------- Top of File ----------
/**
 * Counter.java
 *
 * This file was auto-generated by grid services tooling for the
 *  Globus Toolkit
 *
 */

package org.globus.ogsa.samples.counter;

public class Counter {
    private int value = 0;

    public Counter() {
    }

    public int getValue() {
        return this.value;
    }

    public int add(int value) {
        this.value += value;
        return this.value;
    }

    public int subtract(int value) {
        this.value -= value;
        return this.value;
    }

}
```

Grid Service Description to Grid Service (5 of 5)

- Deploy the gar:
  - igt-deploy-gar $GLOBUS_LOCATION/sample2/topDownCounterService.gar
- Start the container
  - igt-start-container
- The Counter Service is up!
- You will be able to see it in the list of services in Grid Services Manager

Summary

- Tooling can shield developers from the complexity of grid service development
- Tooling can remove many of the complexities by
  - Hiding many of the technologies used in developing grid service code
  - Providing programmatic generation for all parts of a grid service
- Tooling can provide an easy-to-understand interface for grid service developers.
- Tooling can provide a rich set of commands for grid service development.
- For more information see:
  - http://publib.boulder.ibm.com/eserver/
  - http://www.ibm.com/redbooks/

Appendix A - Useful information

- IBM Grid Toolbox V3 for Multiplatforms (IGT) homepage
  - http://www-1.ibm.com/grid/solutions/grid_toolbox.shtml
- IGT electronic downloads and service updates
  - https://www7b.software.ibm.com/dl/grid/grid-p
- eServer online Information Center
  - http://publib.boulder.ibm.com/eserver/
  - Please refer to the online version of information
- Redbooks
  - http://www.redbooks.ibm.com/ (search for “grid”)
  - Grid Computing with the IBM Grid Toolbox SG24-6332
  - Grid Services Programming and Application Enablement SG24-6100

Appendix B

Detailed Installation Information

Install Technology

- Install Shield MultiPlatform (ISMP) Edition 5.01
  - Common wizard and User Interaction across platforms
  - Ease of use
  - Wrappers native install media
    - RPMs on xLinux
    - installPs on AIX
  - Includes JRE for install usage
  - Single Binary distribution

    setuplx.bin is 197MB

Wizard Flow

- Welcome
- License
- “ibmgrid” id:
  - obtain pw, create ibmgrid user id, groupid, add user to group
- Optionally deploy services
  - Information services, program management, data management, service group, policy, cmm
- Confirmation
- Progress
- Finish

Other

- Install is to be done as “root”
- Install time is approximately 15-30 minutes
- Back, Next, Cancel buttons
- No selective install
  - All options are installed
- Installer is not re-entrant
  - Will not run if product is installed
- No program groups, shortcuts, menu items etc.

Linux on x Series RPMs installed

- 9 RPMs (vrm is GM level)
- rpm –qa | grep IBMG
  - IBMGrid-3.0-1
  - IBMGrid-OGSA-3.0-1
  - IBMGrid-AppServer-3.0-1
  - IBMGrid-DataBase-3.0-1
  - IBMGrid-WebApp-3.0-1
  - IBMGrid-OpenJMS-3.0-1
  - IBMGrid-ServiceGroup-3.0-1
  - IBMGrid-CMM-3.0-1
  - IBMGrid-Policy-3.0-1

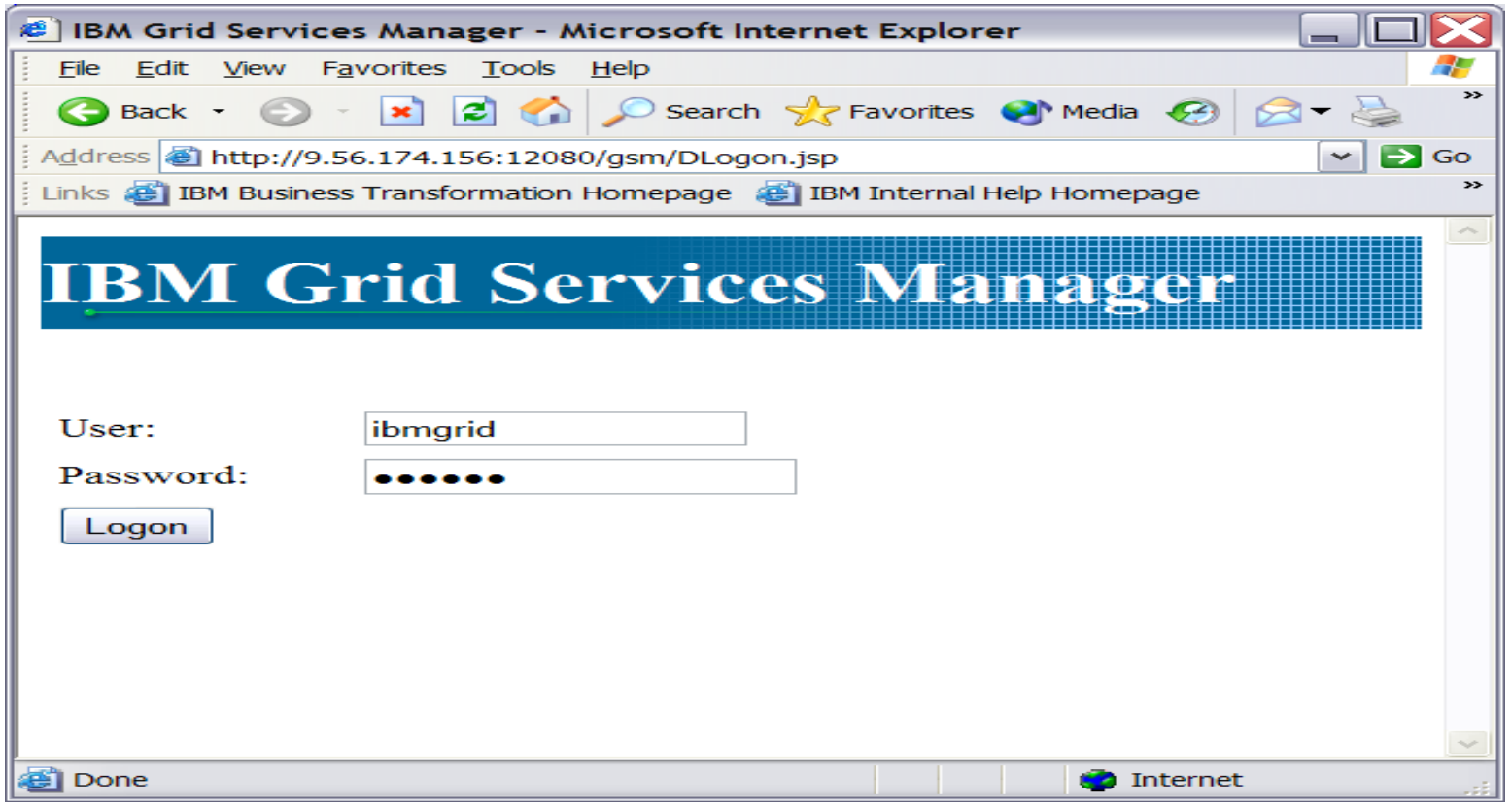
Other install actions

- ISMP:
  - Creates ibmgrid userid and group
  - Installs RPMs
  - Deploys optionally selected services
  - Creates an “ibmgrid” entry in the user registry
  - Maps “ibmgrid” to the administrator role
  - Turns WebSphere security on
- RPM post-install scripts

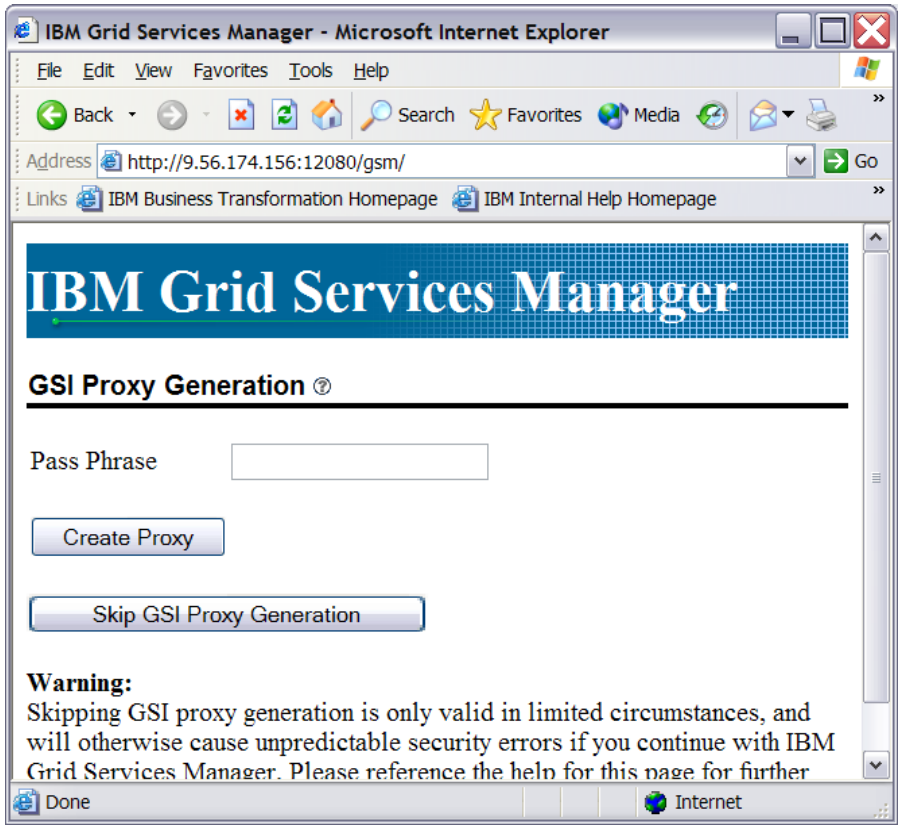
Install Verification

- RPMs are installed
- /opt/IBMGrid directory created (with content)
- ibmgrid user id and group id created
- Start the container
  - Login as “ibmgrid”
  - /opt/IBMGrid/bin/igt-start-container
    - Wait for “server open for eBusiness”
- Point browser to Grid Services Manager web application
  - http://servername:12080/gsm
- At the Logon screen
  - Use “ibmgrid/pw” input during install
  - Would then see a proxy screen (STOP HERE) unless security is configured
- Log files

GSM logon screen



GSM proxy screen



Un-install

- Installer creates an uninstaller
  - You MUST run the uninstaller first
    - /opt/IBMGrid/_uninstall/uninstaller.bin
  - Uninstaller calls native uninstall
    - rpm –e
  - Not all files (other than uninstaller) are removed
    - rm –rf /opt/IBMGrid
  - ibmgrid userid/group are not removed

Re-install

- Requires an uninstall first.
  - As ‘root’ remove all directories
    - rm-rf /opt/IBMGrid/
  - You can optionally remove ibmgrid uid/group
    - If you don’t you need to specifiy the same pw (pw won’t change)
  - Can then do a complete install
  - If security has been set up, certificates are in separate directories (don’t remove them)

Install options

- Logging
  - ./setuplx.bin –log ! @ALL
    - Logfile at: /opt/IBMGrid/log.txt
  - ./setuplx.bin -is:log <logfilename> (ISMP internal tracing)
- Console mode
  - ./setuplx.bin –console
- Silent mode
  - ./setuplx.bin – options-record <respfilename>
  - ./setuplx.bin –options <respfilename> -silent
  - ./setuplx.bin –options-template <templatefilename>

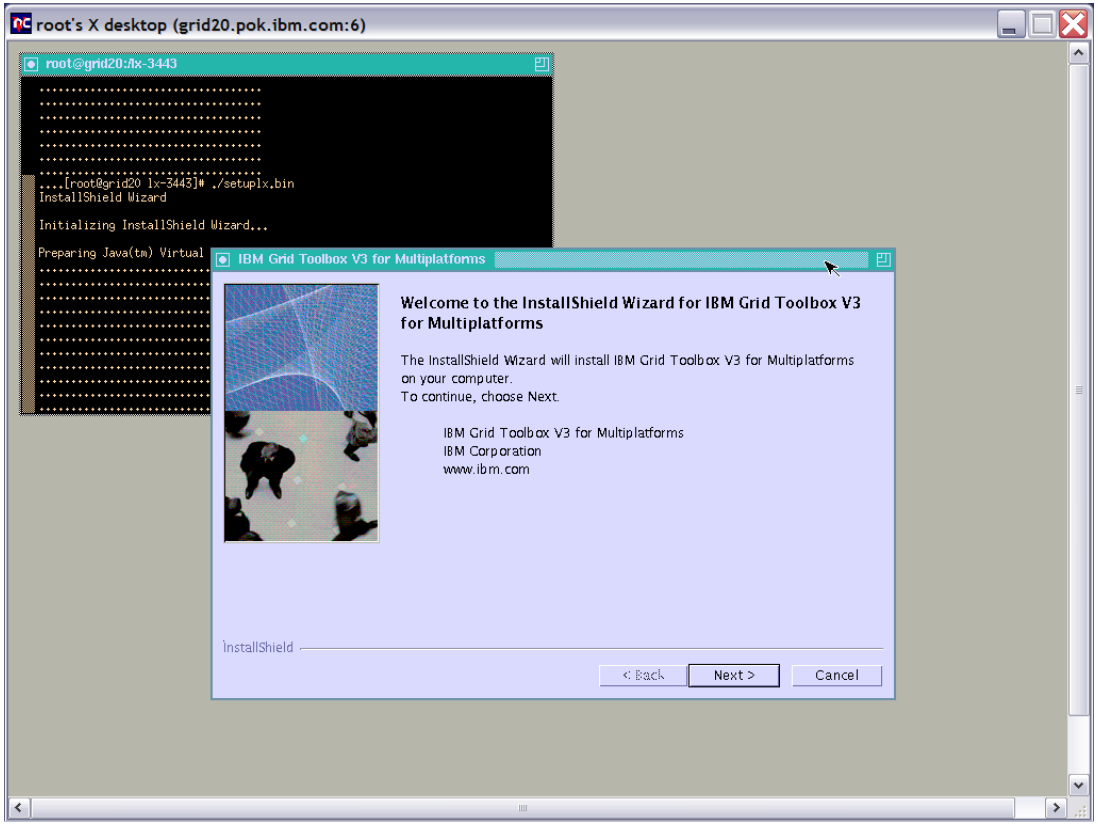
Log files

- /opt/IBMGrid/log.txt (if option on previous page is used)
- /opt/IBMGrid/logs/deploy.txt, deploye.txt, *.*
- /opt/IBMGrid/logs/
  - **omui.log - stdout of mapping of ibmgrid to Administrator role of gsm**
  - **omuie.log - stderr of the above**
  - **deploy.txt - stdout of igt-deploy-gar**
  - **deploye.txt - stderr of igt-deploy-gar**
  - **setperms.log - stdout of the setperms invocation**
  - **setpermse.log - stderr of the above**
  - **start.log - stdout of starting the container**
  - **starte.log - stederr of the above**
  - **stop.log - stdout of stopping the container**
  - **stope.log - stderr of stopping the container**
- /opt/IBMGrid/AppServer/logs/*
- /opt/IBMGrid/gars/deploy.lst

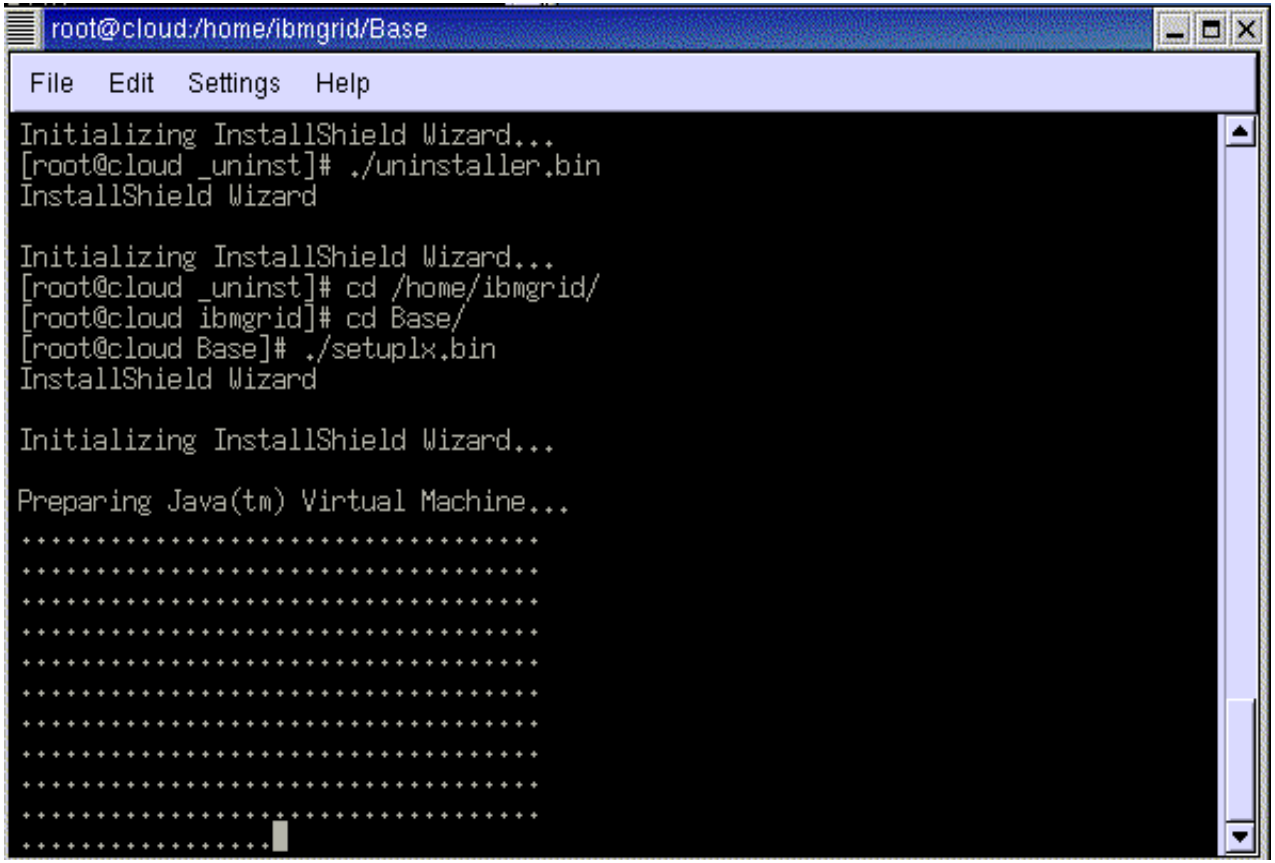
Troubleshooting

- Logs available
- If repeatable, need to uninstall first, then can use the -log option
- Slow machines, RPM timeouts
  - Symptom: Improperly configured Application Server
- User manually removes RPMs
  - rpm –e
    - Symptom: Error when installing (RPM already exists)
    - RPMs have been uninstalled
    - Need to “backup and rename” /root/vpd.properties

Demo


root's X desktop (grid20.pok.ibm.com:6)
root@grid20:/lx-3443
....[root@grid20 lx-3443]# ./setuplx.bin
InstallShield Wizard
Initializing InstallShield Wizard...
Preparing Java(tm) Virtual
IBM Grid Toolbox V3 for Multiplatforms
Welcome to the InstallShield Wizard for IBM Grid Toolbox V3 for Multiplatforms
The InstallShield Wizard will install IBM Grid Toolbox V3 for Multiplatforms on your computer.
To continue, choose Next.
IBM Grid Toolbox V3 for Multiplatforms
IBM Corporation
www.ibm.com
InstallShield
< Back
Next >
Cancel

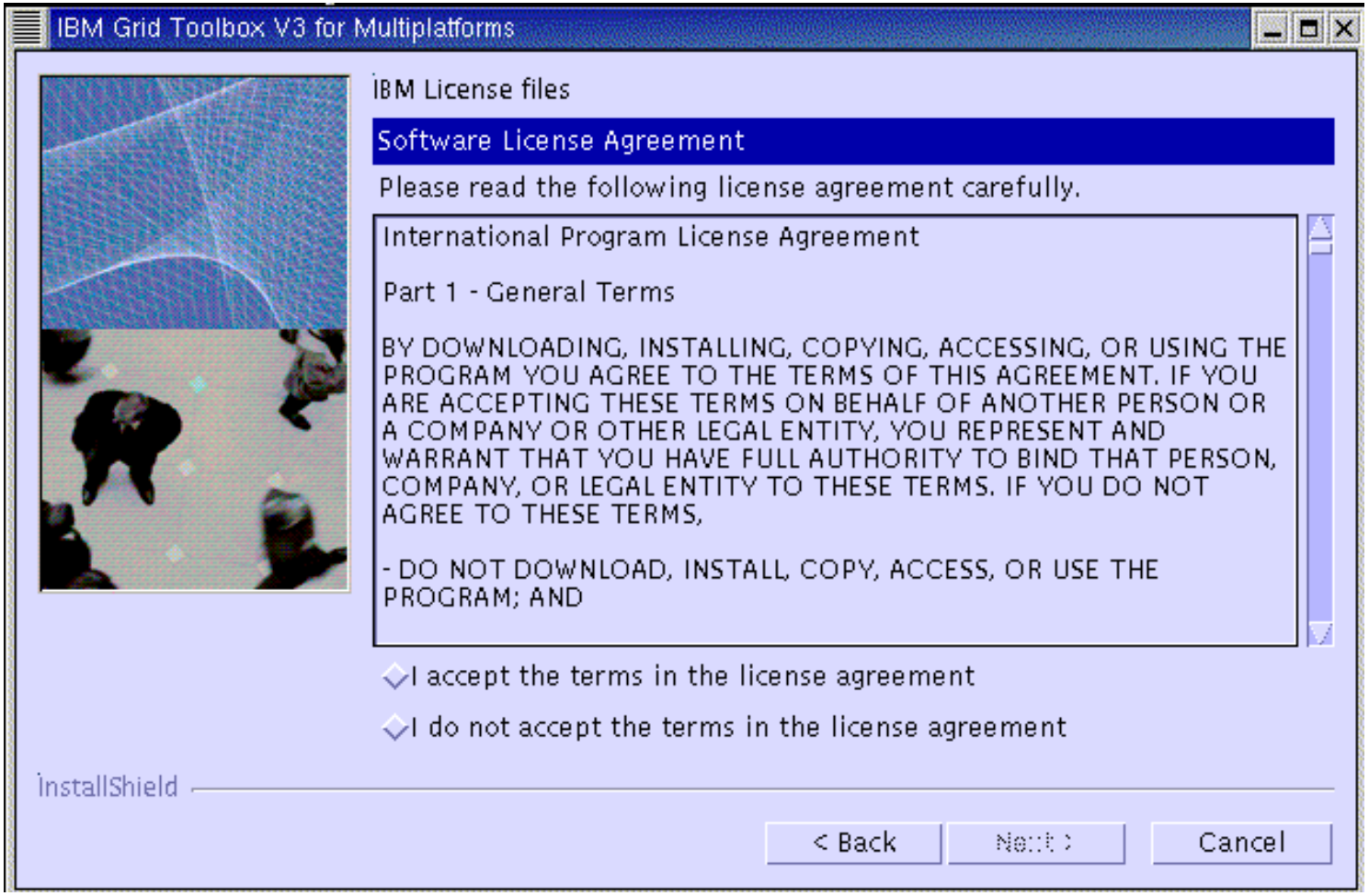
Invoking the installer/uninstaller



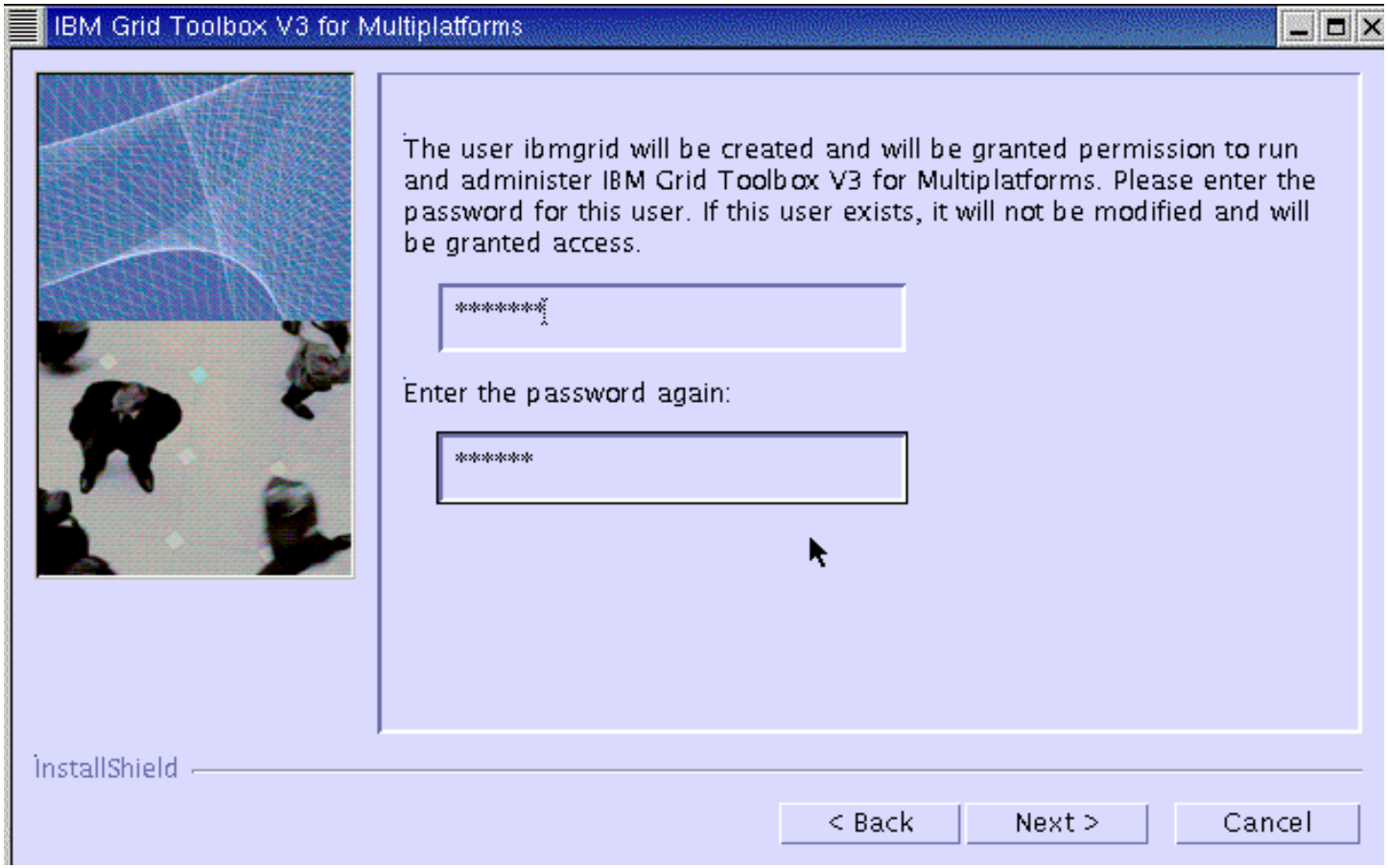
IGT Installer Panels

Welcome Panel

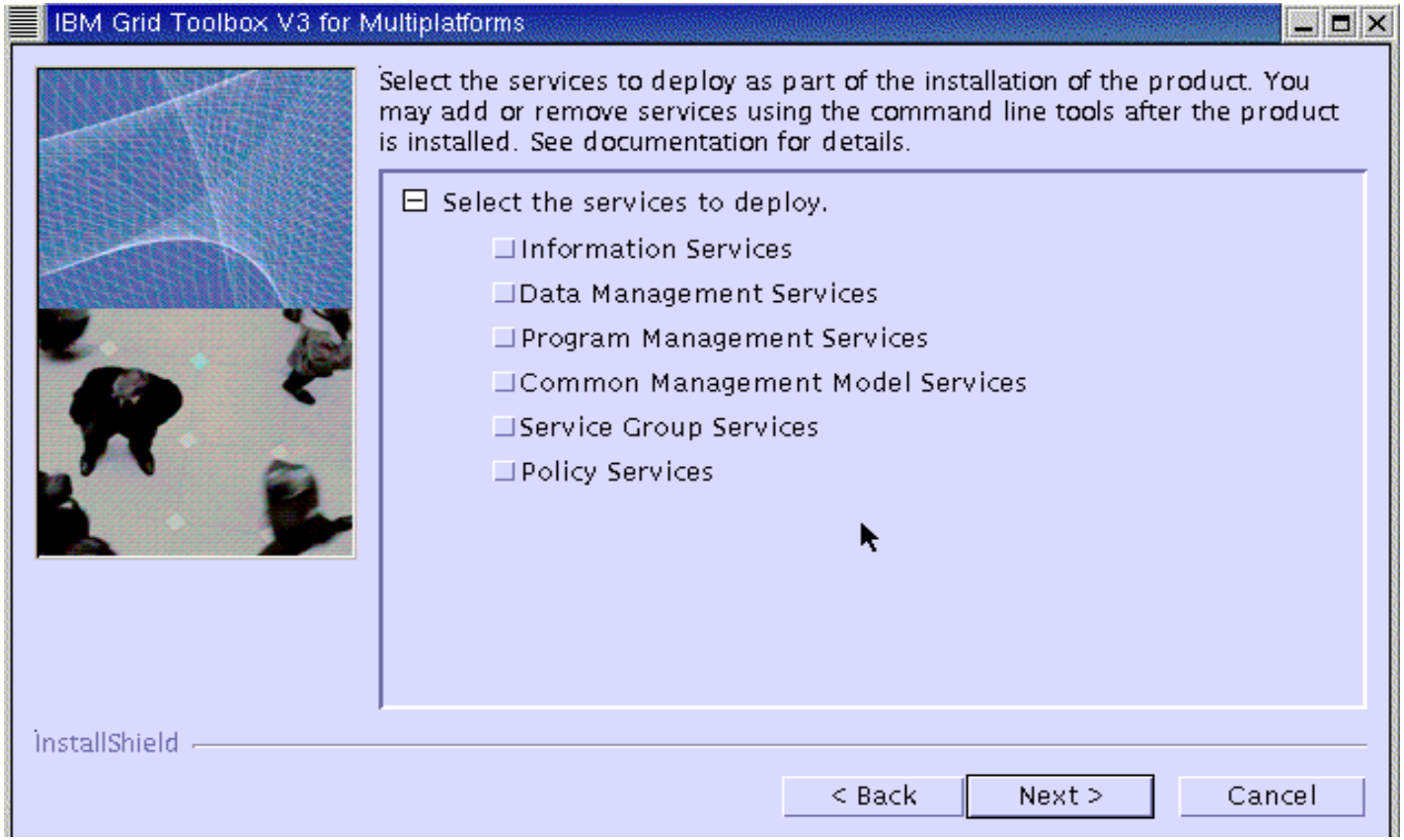
License Panel



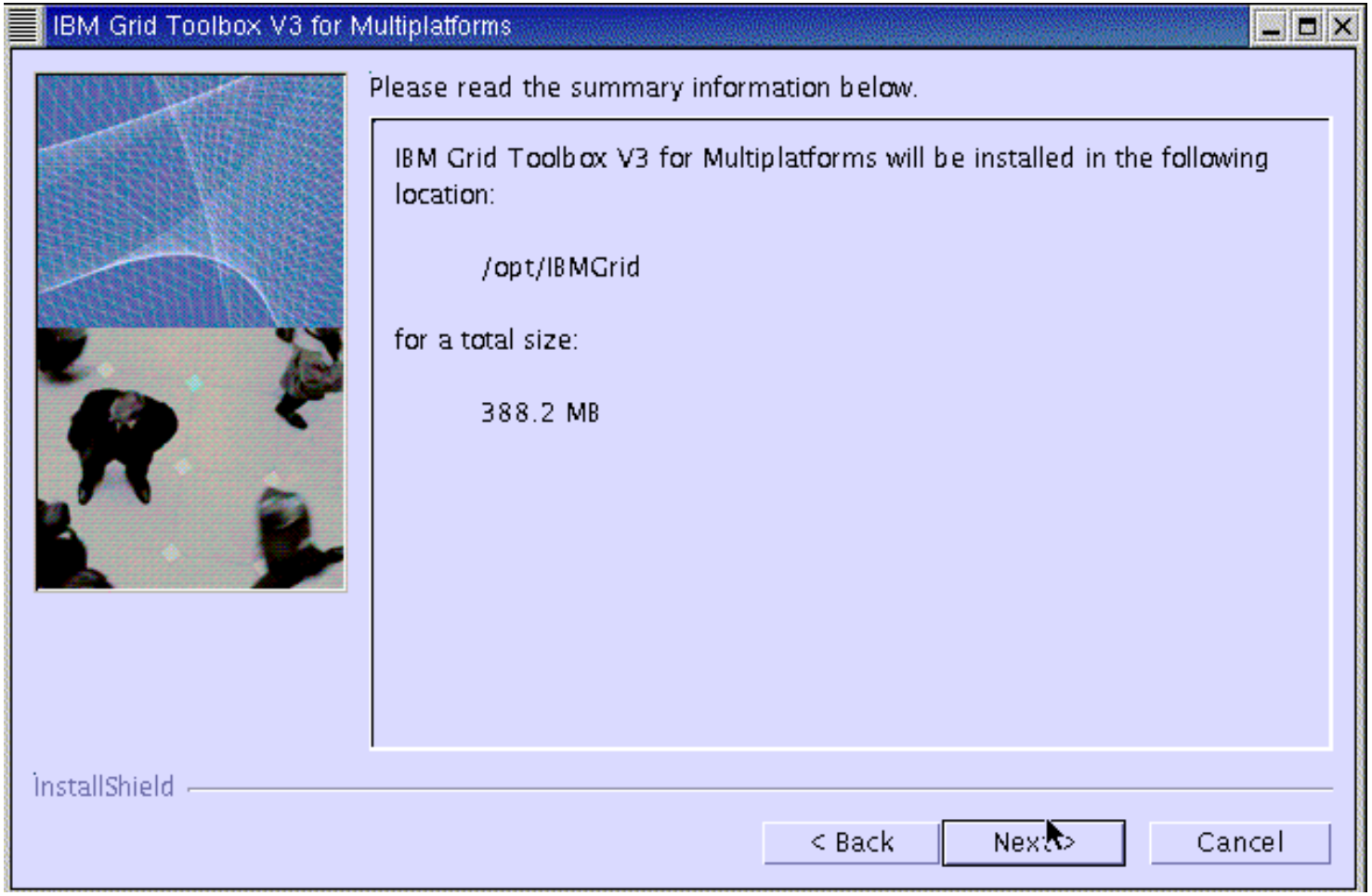
Verify password panel



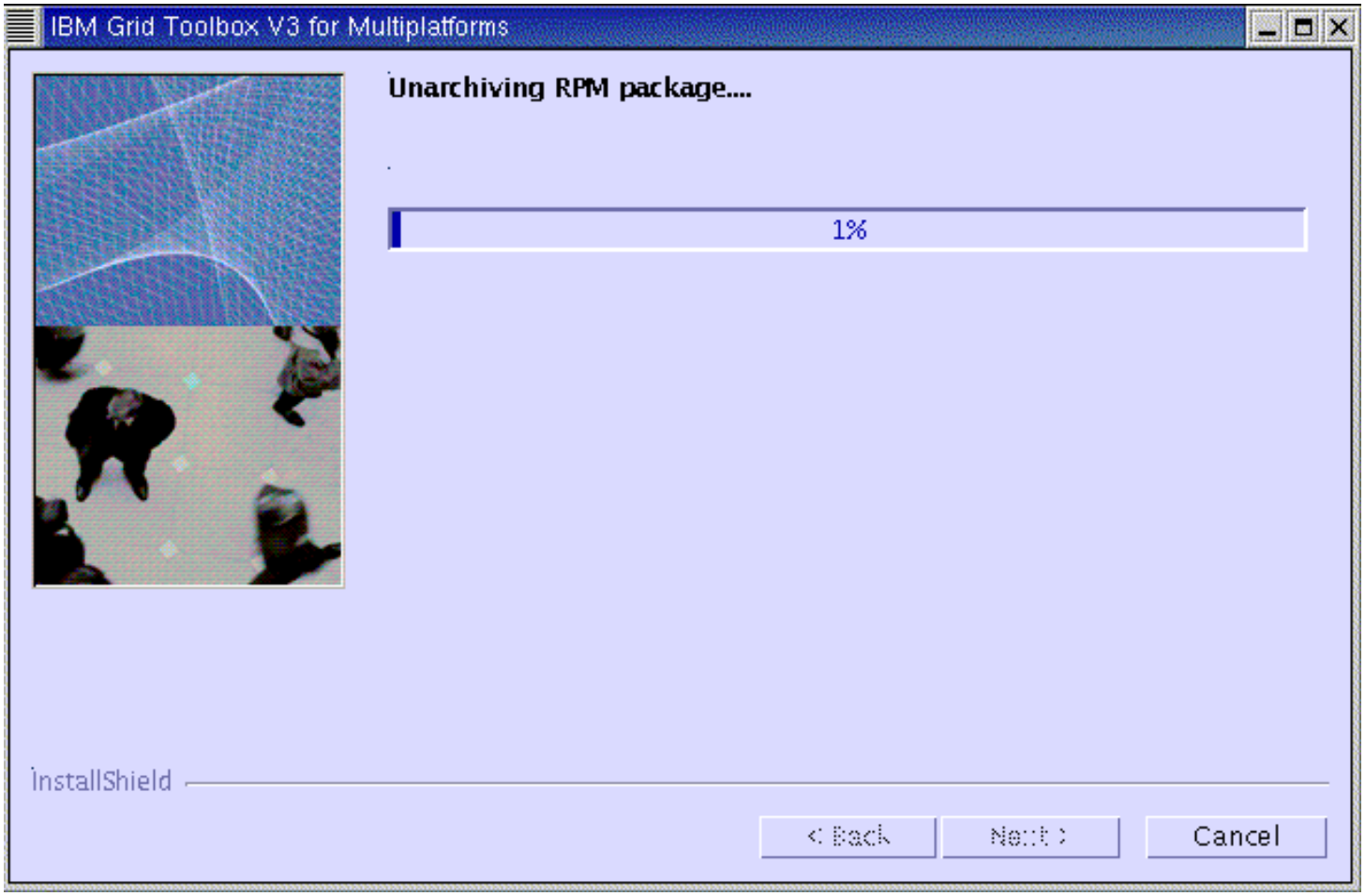
Select services to deploy



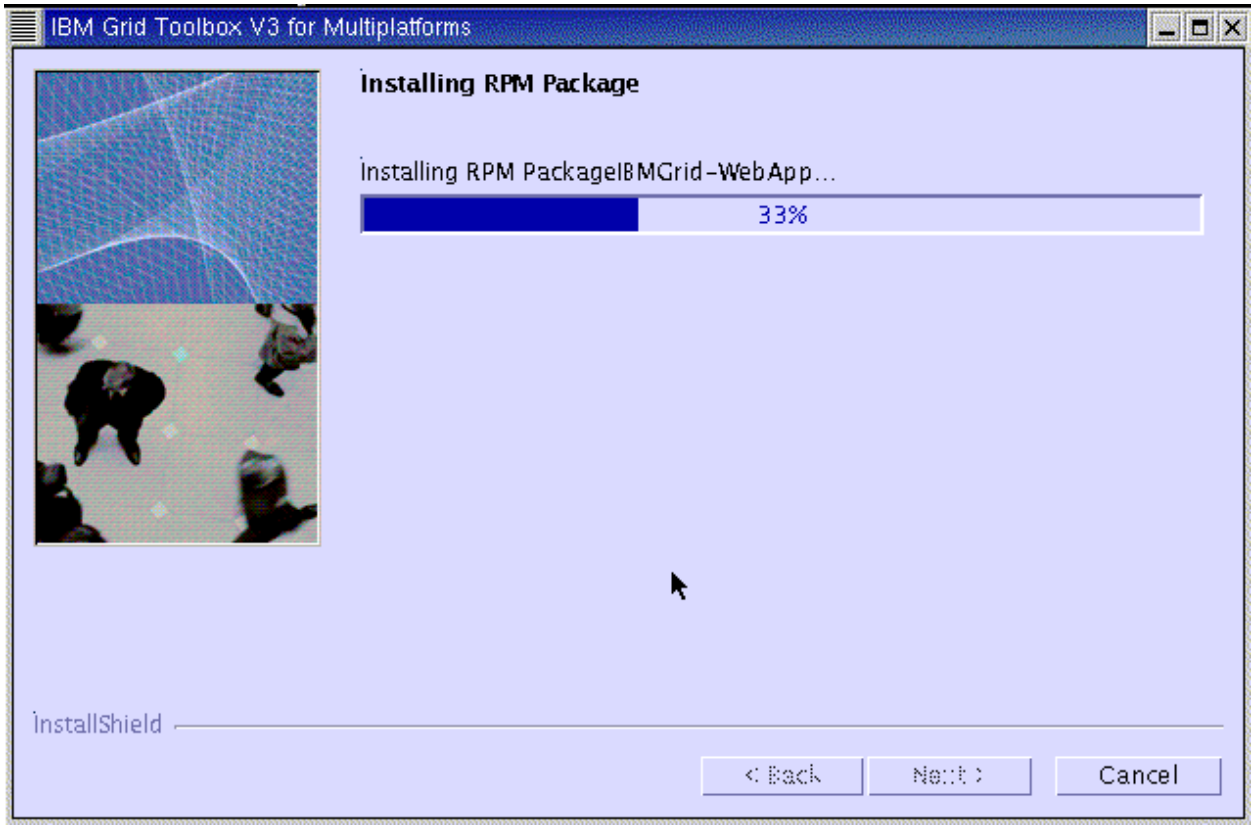
Install Preview Panel



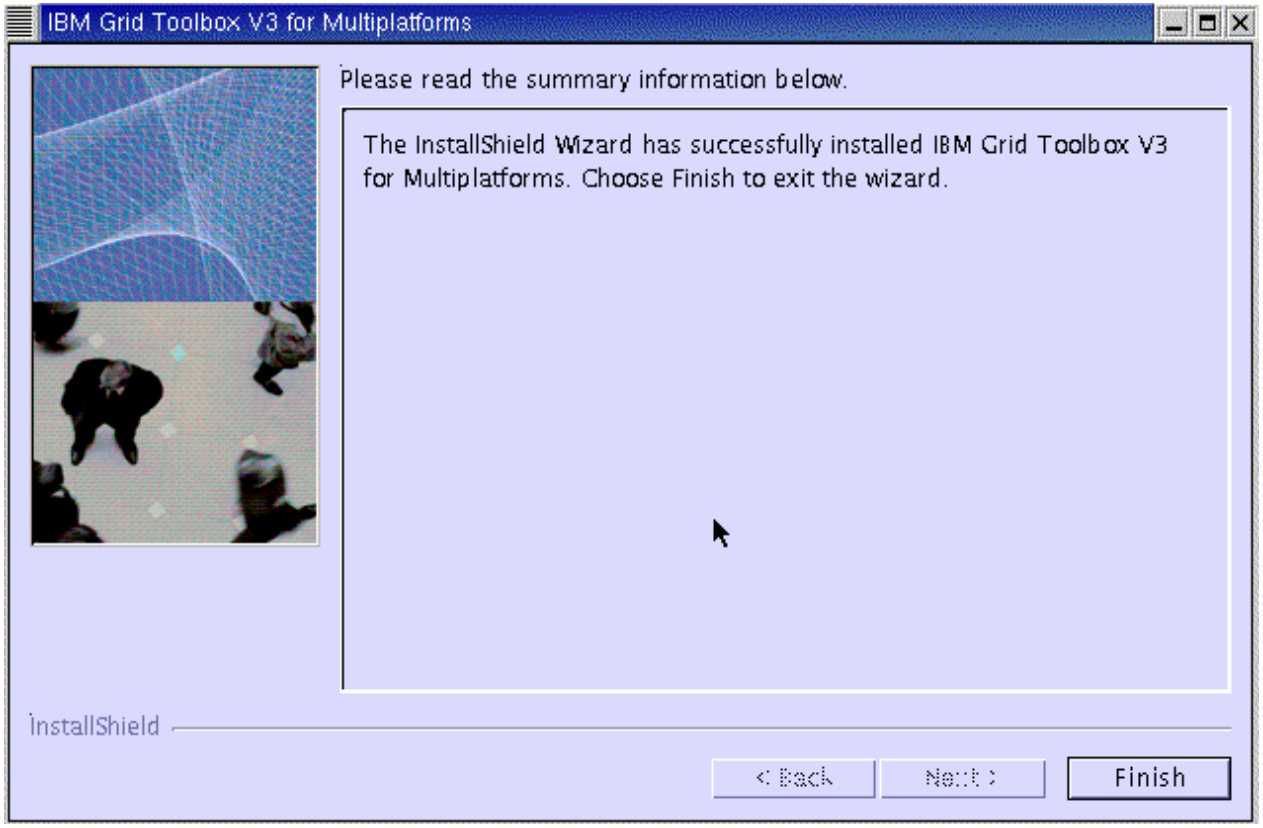
Install Panel



Still installing



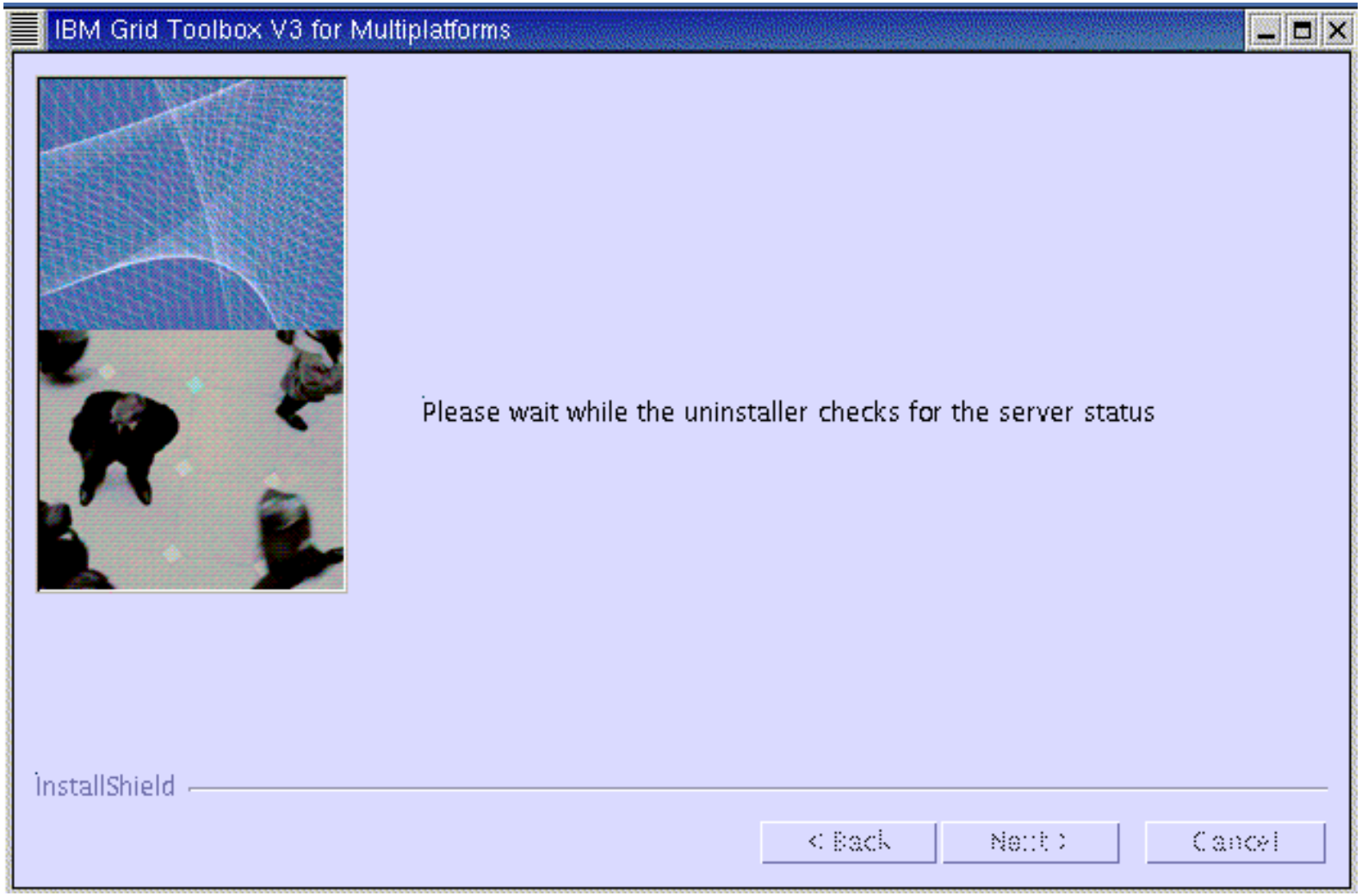
Install Summary Panel



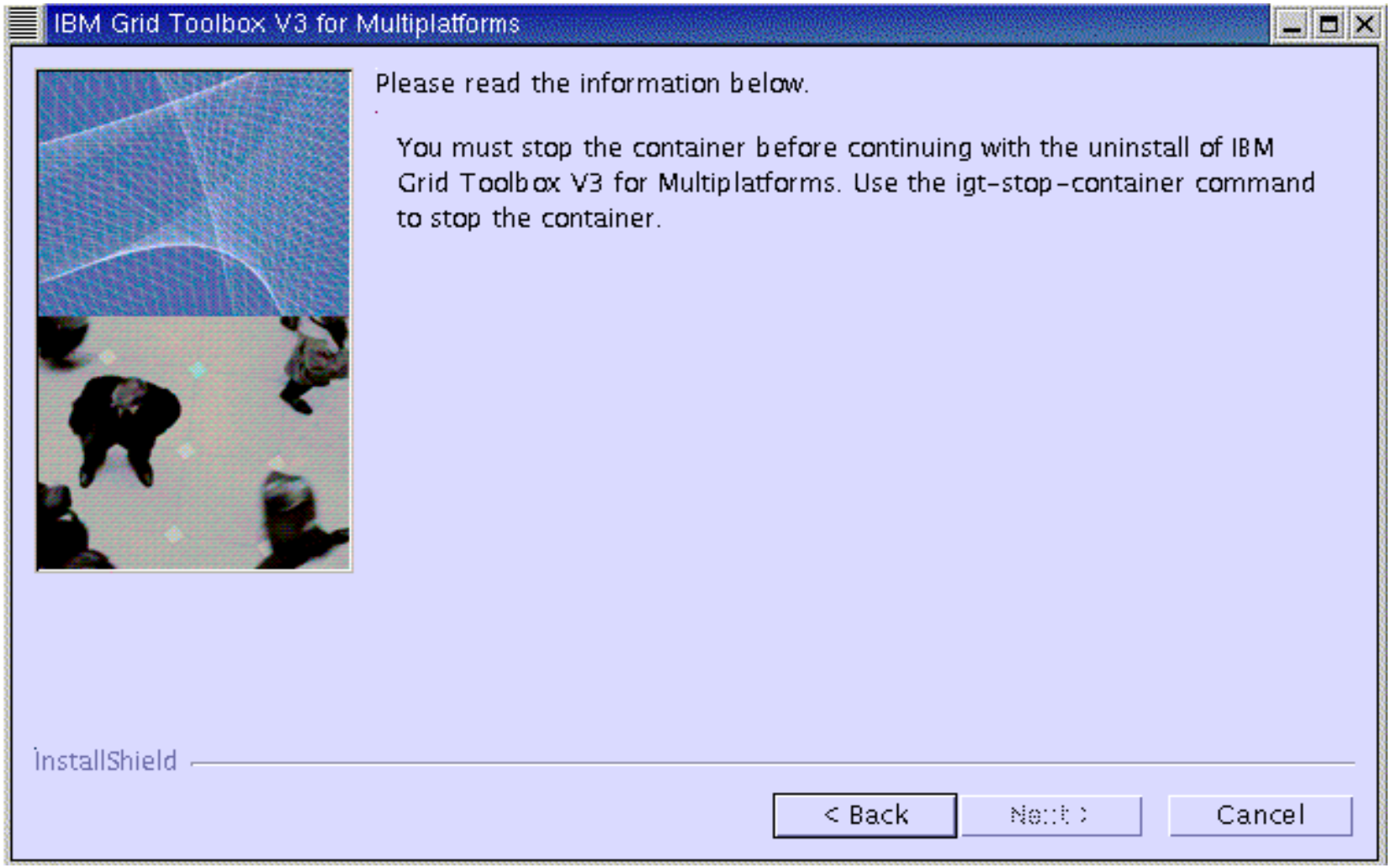
Uninstaller panels

Welcome Panel

Checking if the server is running

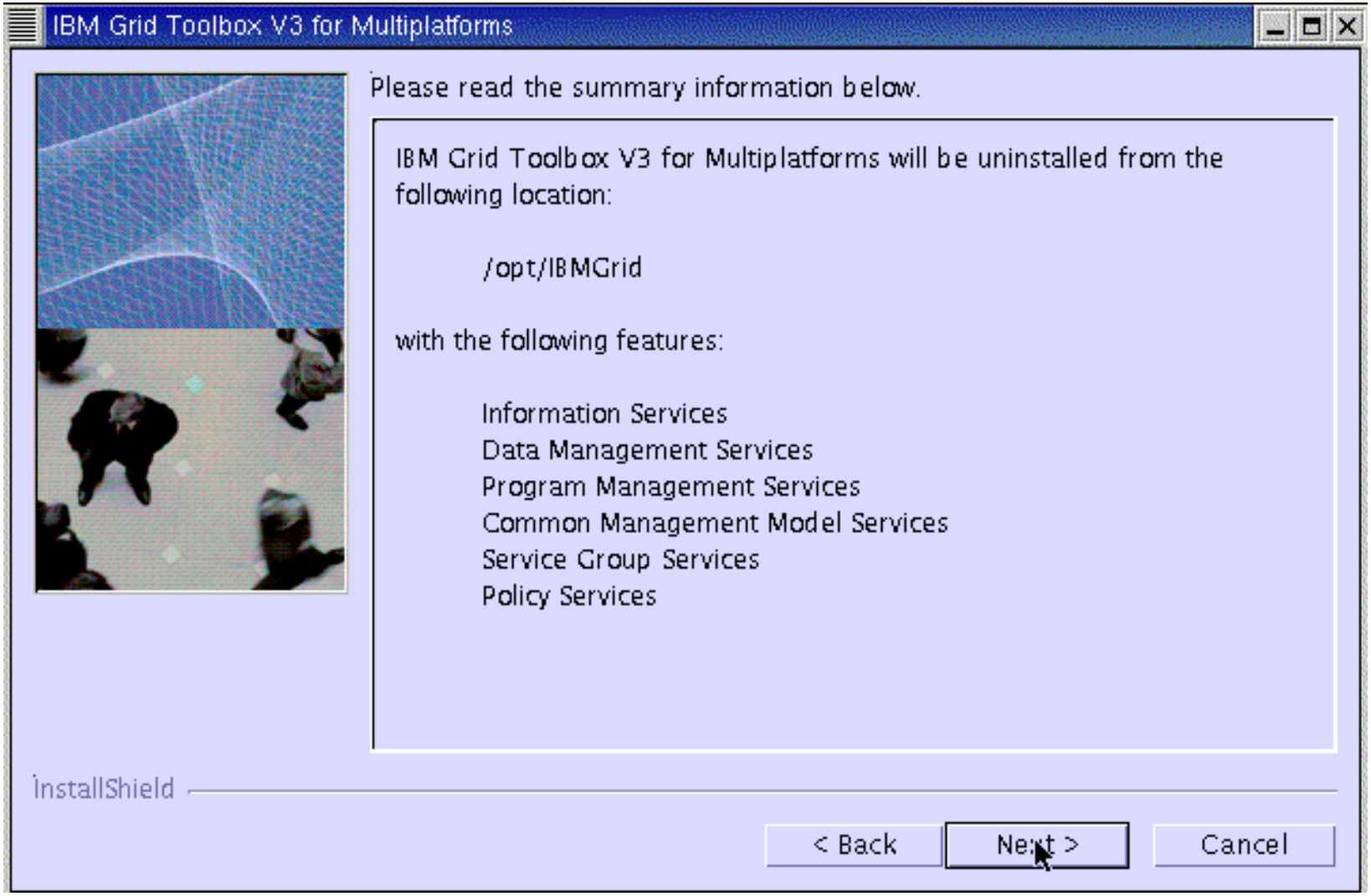


Error message shown if server is running

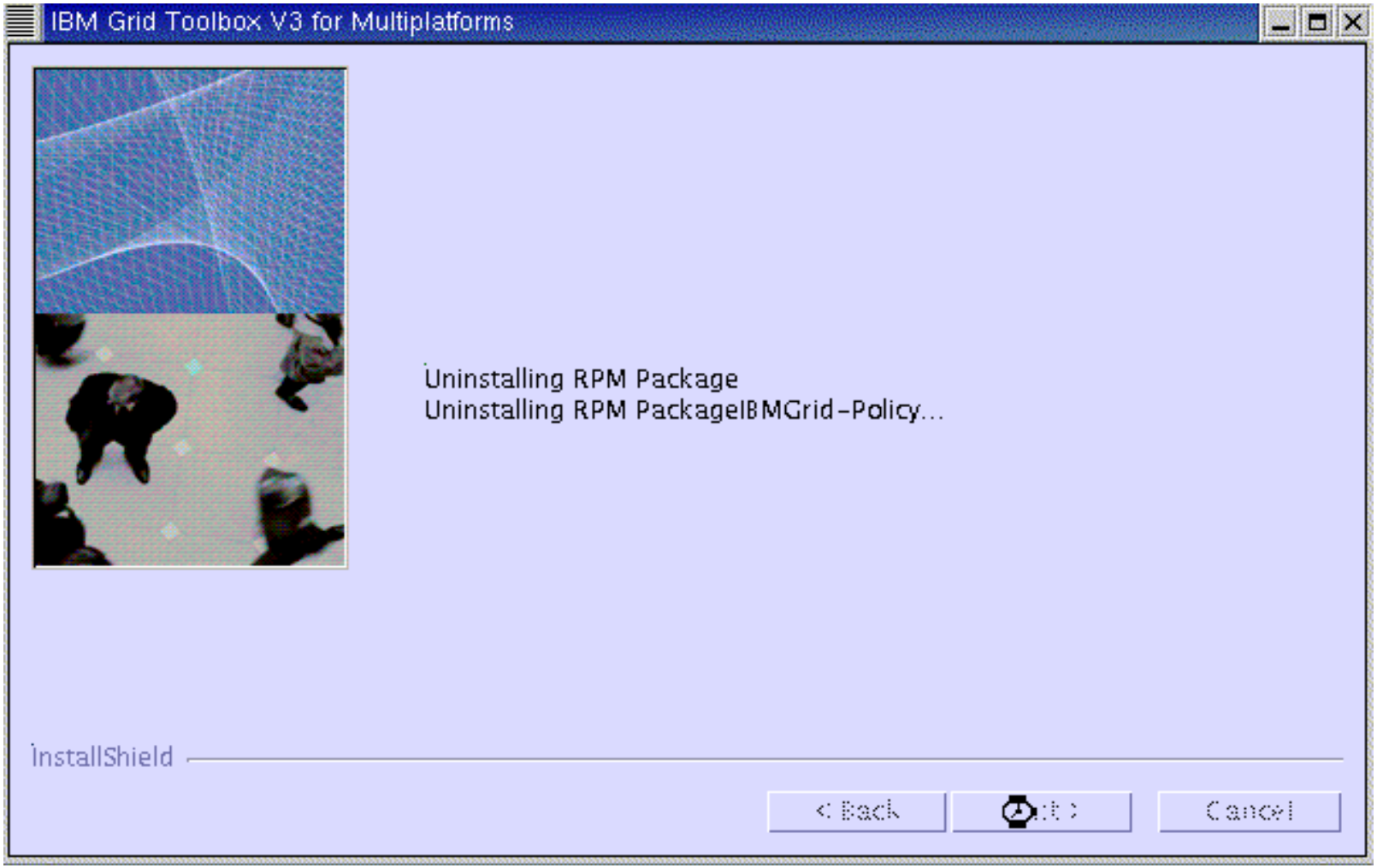
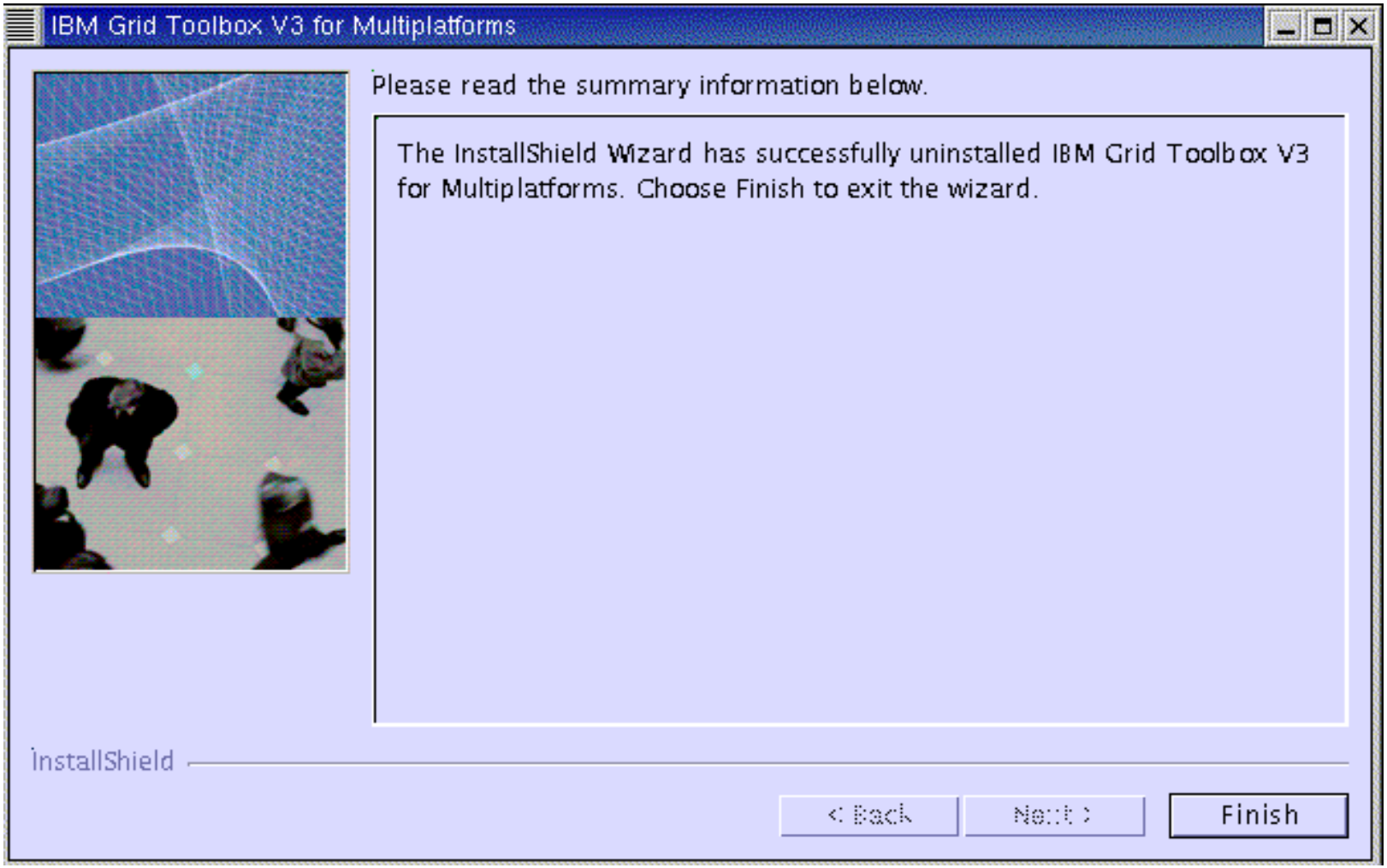


Server not running, showing features to uninstall

Uninstall Preview Panel



Uninstall Panel



Uninstall Summary Panel